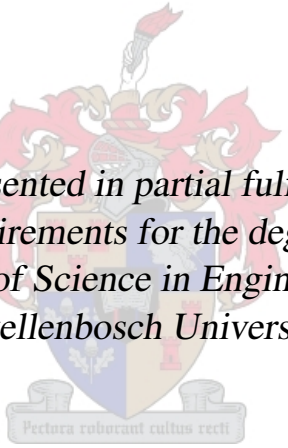
An investigation into PCF-DCF behaviour of 802.11b networks

by

Neville Greyling



Thesis presented in partial fulfilment of
the requirements for the degree of
Master of Science in Engineering
at Stellenbosch University

Supervisor: Dr. Riaan Wolhuter

Department of Electrical and Electronic Engineering

March 2010

Declaration

By submitting this thesis electronically, I declare that the entirety of the work contained therein is my own, original work, that I am the owner of the copyright thereof (unless to the extent explicitly otherwise stated) and that I have not previously in its entirety or in part submitted it for obtaining any qualification.

March 2010

Abstract

In recent years the demand for bandwidth has dramatically increased because of new applications for data and multimedia, and wireless technology has prevailed as a prominent technology for data connectivity, especially for home, office and last mile services.

As wireless communications are dependant upon spectrum availability, which is communal, this scarce commodity in communication has to be used as efficiently as possible. Some aspects of this requirement are addressed in this project.

We chose the IEEE 802.11b standard for this particular investigation because of its widespread use, the vast amount of applicable literature, the variety of software simulation tools and the ease with which equipment can be obtained.

The IEEE 802.11 standard specified the Point Coordination Function as the deterministic protocol. Recently research into this aspect has stagnated, and it was the purpose of this project to investigate how existing infrastructure networks could be improved by optimising some modes of the 802.11 protocol. The investigation also hoped to determine when to change between Distributed Coordination Function (DCF) and Point Coordination Function (PCF), and to provide an adaptive protocol to do so.

This thesis presents mathematical models for the operation of DCF and PCF modes, which is compared with results from a network simulator (*ns*2), for theoretical verification. A protocol is also proposed to dynamically switch between DCF and PCF, to harness the advantages they present.

Opsomming

Die afgelope paar jaar het die aanvraag na bandwydte dramaties verhoog as gevolg van nuwe toepassings vir data en multimedia, en draadlose tegnologie het voorgekom as 'n dominante tegnologie vir data konnektiwiteit, veral vir die huis, kantoor en laaste myl dienste.

Omdat draadlose kommunikasie afhanklik is van spektrum beskikbaarheid, wat gemeenskaplik is, moet hierdie skaars kommoditeit in kommunikasie so effektief moontlik gebruik word. Sekere aspekte van die vereiste sal in die tesis ondersoek word.

Dit is besluit om die IEEE 802.11b standard vir die spesifieke ondersoek te gebruik as gevolg van die wye toepassing, die groot hoeveelheid beskikbare literatuur, die verskeidenheid simulasie sagteware en die gemak waarmee die toerusting bekom kan word.

Die IEEE 802.11 standaard spesifiseer the Punt Koordinasie Funksie (PCF) as die deterministiese protokol vir die betrokke standaard. Onlangs het navorsing oor hierdie aspek gestagneer, en dit is die doel van die projek om te ondersoek hoe bestaande infrastruktuur netwerke moontlik verbeter kan word deur optimering van sekere modusse van die 802.11 protokol. Die ondersoek hoop ook om te bepaal wanneer die oorgang van die Distrubusie Koordinasie Funksie (DCF) en Punt Koordinasie Funksie sal plaasvind, en om 'n dienooreenstemmende protokol te ontwikkel.

Die tesis verskaf wiskundige modelle vir die werking van die DCF en PCF modusse, wat vergelyk word met resultate uit 'n netwerk simulator (*ns*2), vir teoretiese verifikasie. 'n Protokol word ook voorgestel om dinamies te wissel tussen DCF and PCF, om die voordele wat die protokolle verskaf te gebruik.

Acknowledgements

I would like to express my sincere gratitude to the following people, without them this thesis would not have been possible. My study leader, Dr Riaan Wolhuter, for his guidance, advice, wisdom, time, enthusiasm, encouragement and calming affect in tough times. My girlfriend Erin for a full tummy, love and endless nights of typing. My parents, what can I say. Our creator.

Contents

List of Figures

List of Tables

Nomenclature

Abbreviations

ACK	Acknowledgement
AP	Access Point
ARP	Address Resolution Protocol
BS	Base Station
BSS	Basic Service Set
CDMA	Code Division Multiple Access
CFP	Contention Free Period
CP	Contention Period
CRC	Cyclic Redundancy Check
CTS	Clear To Send
DCF	Distributed Coordination Function
DS-CDMA	Direct Sequence-Code Division Multiple Access
DSDV	Destination Sequence Distance Vector
DTIM	Delivery Traffic Indication Map
ESS	Extended Service Set
FDMA	Frequency Division Multiple Access
FH-CDMA	Frequency Hop-Code Division Multiple Access
FIFO	First In First Out
GUI	Graphical User Interface
IBSS	Independent Basic Service Set
LIFO	Last In Last Out
LAN	Local Area Network
LL	Link Layer
NAV	Network Allocation Vector
PC	Point Coordinator
PCF	Point Coordination Function
PDF	Probability Distribution Function
pmf	probability mass function
RF	Radio Frequency

RRP	Round-Robin Polling
RTS	Request To Send
TCP	Transfer Control Protocol
TIM	Traffic Indication Map
TU	Time Unit
QoS	Quality of Service
TDMA	Time Division Multiple Access
UDP	User Datagram Protocol
VCS	Virtual Carrier Sense
WLAN	Wireless Local Area Network

Prefixes

f	femto	10^{-15}
p	pico	10^{-12}
n	nano	10^{-9}
μ	micro	10^{-6}
m	mili	10^{-3}
d	deci	10^{1}
k	kilo	10^{3}
M	mega	10^{6}
T	tera	10^{9}

Units

b	bit
B	Byte
Hz	Hertz
m	Meters
min	Minutes
s	Seconds
V	Volt
W	Watt

Chapter 1

Introduction

1.1 Background

In recent years the demand for bandwidth has dramatically increased because of new applications for data and multimedia. These applications would include web browsing, social networking, e-mail, voice over IP, video streaming and many more. This has led to an evolution of innovation, in ways to provide connectivity to as great an audience as possible. A research area that has especially enjoyed attention is wireless technology.

Wireless technology has prevailed as a prominent technology for data connectivity, especially for home, office and last mile services. Some of its strong points are low cost, ease of implementation and the new features it provides with mobility, as compared to wired technology. Another major advantage of this technology, given that the network is not overloaded and taking signal range considerations into account, is the ease with which new nodes can be added without the need for physical wiring. Unfortunately its range is limited due to factors such as free space loss, scattering, multipath and absorption. Also, to compensate for these losses, additional overhead is required that, in turn, reduces the effective throughput of such a network.

As wireless communications are dependant upon spectrum availability, which is communal, this scarce commodity in communication has to be used as efficiently as possible. Some aspects of this requirement are addressed in this project.

Typical wireless technology for data connectivity in use today would be all variations of the IEEE 802.11 (wifi) standard, all variations of the IEEE 802.16 (WiMaX) standard, and all the G standards used by cellular operators.

We chose the IEEE 802.11 standard for this particular investigation because of its widespread use, the vast amount of applicable literature, the variety of software

simulation tools and the ease with which equipment can be obtained. Some principles of this project could possibly apply to other wireless standards.

The physical layer of the IEEE 802.11 standard, which represents the hardware and interconnections, has a fixed structure and not much can be done to improve this, especially in existing networks. On the other hand, the media access control (MAC) layer which is responsible for the construction of packets, and the control of how the physical layer is managed, warrants further research. Thus all work will be done on the MAC layer.

The IEEE 802.11 standard has provided two protocols for operation in the MAC layer, namely the Point Coordination Function (PCF) and Distributed Coordination Function (DCF). DCF is a contention based protocol and the most widely used method for packet communication in wifi. It relies on the Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) protocol and is generally used in ad hoc networks. PCF on the other hand, is a deterministic protocol. Both these two modes of operation have their strengths and weaknesses and warrant further investigation.

1.2 Objectives

The objectives of this thesis are:

- To make an in depth study of the IEEE 802.11 MAC layer.
- To identify to which situations PCF and DCF are respectively better adapted.
- To simulate a typical IEEE 802.11 network with a suitable simulation package, with the aim to investigate system characteristics such as throughput, packet delays and typical buffer occupancy.
- To identify, and possibly adapt, mathematical models to which all results can be compared for theoretical verification.
- To create a combined PCF and DCF adaptive algorithm to optimize efficiency of an IEEE 802.11 network.

The following are not the objectives of this thesis:

- To study the effects of different IEEE 802.11 configuration parameters, such as contention window sizes, retry limits, slot times or inter frame spaces.
- To address any problems related to determining optimum values for the Super Frame (SF) and the Contention Free Period Maximum duration (CFPMaxdur).

- To address any problems related to ad-hoc networks.
- To address any issues concerning different 802.11 standards.
- To address any problems regarding noisy channels and signal loss.

1.3 Design constrictions

Due to research on the IEEE 802.11 standard being so open-ended and broad, some restrictions had to be put in place to narrow down the engineering problem to a manageable size. The first design constraint chosen was that the system would operate in an infrastructure mode, implying that all wireless nodes are stationary and will communicate through a central base station called an Access Point (AP). Only the stations communicating with the AP, and not the AP communicating with the nodes, in our simulations and mathematical model will transmit data packets. That is, the AP will transmit all control frames to the nodes, but not any data packets. Another design constraint chosen for mathematical modelling was that all stations will transmit at a fixed maximum data rate and fixed packet sizes.

1.4 Approach

A general background is required of communication protocols and their evolution for IEEE 802.11 standardization to provide further insight into the engineering problem. Relevant work on the IEEE 802.11 standard was studied, to ensure that an accurate reference frame was used for design and implementation considerations. With a basic system outline in mind, a simulation platform was identified for protocol development and system testing. The platform, if not exactly sufficient for the problem, had to be adapted accordingly. Design constrictions and predefined criteria were chosen to narrow the engineering problem down a manageable size; mathematical models had to be identified in agreement with this. After testing and the comparison of simulation and mathematical results, it was possible to identify under the circumstances under which each protocol is best suited. This led to developing a protocol that would harness the strengths of both DCF and PCF operation modes.

1.5 Overview of the thesis

The rest of the thesis is set out as follows:

Chapter 2

Before any investigation can be made into the working of the IEEE 802.11 standard it is necessary to investigate the principles on which it is based. This chapter

investigates various existing multi-user single-channel protocols, summarizes the merits of each and identifies the protocols which have application to 802.11 standard.

Chapter 3

This chapter provides an overview as well as an in depth investigation into the operation of the IEEE 802.11 standard. The purpose is to identify and to take into consideration all parameters of importance which will influence the results. A lot of the network terminology used in the project will be found in this chapter.

Chapter 4

This chapter signifies the importance of a software simulation tool to model an IEEE 802.11 network which operates with PCF nodes. The criteria used for the identification of the simulation tool are described and, accordingly, *ns*2 was decided on, based on a PCF toolbox which was available. The configuration parameters, applicability and the shortcomings of the simulator were investigated here.

Chapter 5

This chapter provides the experimental procedures applicable to configuring an IEEE 802.11 b network in *ns*2. All parameter specifications and design considerations are included. Changes made to the existing code to obtain statistics and to ensure that PCF nodes are only operating in the CFP are described. The results in this chapter are compared to the basic multiple access protocols on which the b standard is based, and this chapter serves as a basis to which all mathematical modeling will be compared. General observations of the results obtained are also provided.

Chapter 6

An investigation into existing mathematical models to model the DCF behaviour will be found in this chapter. The performance metrics that were investigated include the throughput, packet delay and the buffer occupancy. All the mathematical results are quantified and compared to the results obtained by simulation.

Chapter 7

This chapter will discuss the investigation of existing mathematical models to model the PCF behavior of an IEEE 802.11 b network. The results obtained are quantified and compared to the simulation model. The performance metrics are investigated, as are the throughput, packet delay and buffer occupancy.

Chapter 8

The mathematical models of Chapters 8 and 9 and the results obtained are combined for the purpose of this project to model the complete behaviour of a network incorporating PCF. No existing method did this; therefore the methods devised for the purpose of this project will be described.

Chapter 9

In this chapter an investigation into the circumstances for which PCF and DCF are better suited when they are incorporated in an infrastructure network is presented. A proposed protocol is also presented to harness the advantages of both protocols.

Chapter 10

This chapter provides the conclusions that can be drawn from the thesis and suggests areas for further research

List of references

Appendix A

A data sheet of a typical IEEE 802.11 b node is included to prove that some design considerations are firmly based on practise.

1.6 Summary and contributions

The work completed during this investigation and associated contributions, can be summarised as follows:

- An extensive investigation into which mode, DCF or PCF, is best suited under different traffic scenarios.
- An extensive working simulation in a software simulator.
- A combined mathematical model for both PCF and DCF that will determine the expected behaviour of the system and its statistical characteristics. The inclusion of packet delay and buffer occupancy are particular additions to event models.
- An algorithm to dynamically adjust between the DCF and PCF modes for optimal operation.

Chapter 2

Overview of protocols

In order for two (or more) data communication nodes to transmit data via the same channel a consensus is necessary between them as to when which node is allowed to access the channel. It becomes crucially important when the number of nodes, or offered data loads, increases. All nodes have to adhere to a set of stipulated rules known as the data transfer protocol.

In order to obtain a better understanding of the IEEE 802.11 a/b/g standards it is imperative to investigate the necessary data transfer protocols. This chapter examines the existing protocols in use and serves as an overview of the protocols that were chosen for use.

2.1 OSI-7 model

In an attempt to standardise protocols and where they fit into a system, the International Standards Organisation (OSI) has specified a seven layer model. The layers are:

1. Physical layer - The hardware used, and also modulation and demodulation
2. Data link layer (or Medium Access Layer) - Manages data transmission and error control.
3. Network layer - Performs routing and congestion control.
4. Transport layer - Establishes connections across the network.
5. Session layer - A type of user interface for the transport layer. Exercises dialogue control and token management.
6. Presentation layer - Syntax and semantics which govern the format in which information is transmitted and presented.

7. Application layer - Interface for the end user. Supports the end-user application process, which includes file formatting and terminal compatibility.

It is important to note that certain protocols may cover more than one layer of this protocol stack. More detail about the various layers can be found in [Black-93]. The focus of this project will be on the Medium Access (MAC) layer.

2.2 Basic communication principles

Basic communication principles can be broadly categorised as follows.

2.2.1 Channel Access Mechanisms

Simplex - communication is in one direction only, e.g. television

Half-Duplex - Communications in both directions, but not simultaneously, e.g. a walkie-talkie

Full-Duplex - Communication in both directions simultaneously, e.g. telephones

Communication in an IEEE 802.11 network is half-duplex. A station cannot listen to the channel and transmit packets at the same time.

2.2.2 Timing

Synchronous transmission - All packets have the same frame size, and the same length header and footer, with the address and data in between. The header is used for the receiver to synchronise with the transmitter.

Asynchronous transmission - Variable packet sizes are used. The packets consist of a start bit, header,a fixed or variable length of data, error control bits (such as a CRC) and one or more stop bits.

The IEEE 802.11 standard uses asynchronous transmission principles.

2.3 Multiple Access Protocols

These types of protocols are concerned with a single data channel accommodating multiple users. Protocols of this type are known as multiple access protocols. They can be divided in three broad categories, namely fixed assignment, centrally scheduled and contention protocols. In sections 2.4, 2.5 and 2.6 these protocols will be examined with the IEEE 802.11 a/b/g standards in mind. Note that this is not an exhaustive list. Several protocols that are not applicable were omitted.

2.4 Fixed Assignment Protocols

2.4.1 TDMA (Time Division Multiple Access)

Note that TDMA is not applicable to the IEEE 802.11 standard. It is sufficient to note that in this method each node is assigned a time slot in which transmission can take place.

2.4.2 FDMA (Frequency Division Multiple Access)

Note that FDMA is not applicable to IEEE 802.11 standard. It is sufficient to note that in this method the entire bandwidth available for transmission is subdivided into sub-bands and each node is assigned one of these sub-bands or a combination to transmit in.

2.4.3 CDMA (Code Division Multiple Access)

In CDMA stations transmit continuously over the same frequency. Each station combines its data with a binary signature, or *code*, which is unique to that station. The receiving station carries knowledge of these codes and is able to use it to distinguish the different transmissions from one another. The set of code must fulfil the following requirements:

1. be easily distinguishable from a replica of itself in time
2. be easily distinguishable, regardless of other codes used on the network.

The combination of the code with the useful data results in a signal requiring more bandwidth than the data alone, and which is referred to as a spread spectrum technique.

The two main types of CDMA which are used are Direct Sequence (DS-CDMA) and Frequency Hopping (FH-CDMA). See [Maral-99, pp. 172-184] for a comprehensive explanation.

Advantages

- Simple to operate as no synchronisation is necessary
- Offers useful properties of protection against interference
- Whole bandwidth available for all users

Disadvantages

- Larger bandwidth is required
- Difficulty of implementation

2.5 Centrally Scheduled (Deterministic) Protocols

Centrally scheduled protocols generally appoint one of the nodes as the server which controls the communication on the channel. The server for the IEEE 802.11 a/b/g standard is referred to as an Access Point (AP) and is also known as the Base Station (BS).

2.5.1 Round-Robin Polling

In RRP each node is on a polling list as the server sequentially cycles through the list in turn and enquires of each node by method of a poll packet. If the node has a packet to transmit it replies with the data packet, and otherwise the server may wait for a predefined timeout period or the station will reply with a "no data packet", before polling the next node on the list.

This is the simplest form of polling and easiest to implement. Note that the time between transmissions for a particular station can become lenghty if the number of stations present is large. RRP can also be inefficient in low traffic scenarios, as a substantial amount of bandwidth is wasted on polling stations with no data.

There are many variations on polling; consult [Tobag-80] for a further review. Here are but a few:

Adaptive Polling

Groups are polled as a whole to determine their traffic demand. If the traffic demand exceeds the initial group poll, the nodes are polled in smaller and smaller groups until all the data has been processed. This format basically follows that of a binary tree structure.

Priority Polling

Stations with higher input data rates are identified in advance and simply polled more frequently.

Reservation Round Robin Polling

Reservation RRP uses polls to determine which nodes have data to transmit during a reservation cycle. Nodes with data reply to those polls to reserve a slot for transmission. During this time nodes that have reserved a slot transmit. This is generally, but not necessarily, a slotted system.

2.5.2 Summary of Round Robin Polling

Round Robin polling generally has the following attributes:

Advantages

- Completely stable
- No bandwidth wasted due to collisions
- Is deterministic in nature, resulting in easy implementation and throughput prediction
- Simple to implement
- Efficient at higher traffic rates than contention protocols

Disadvantages

- Long propagation delays result in inefficiency
- A large number of stations can introduce long delays
- Less efficient at lower packet loads than contention protocols

2.6 Contention Protocols

The widespread use and implementation into various fields of these protocol makes it imperative to investigate them further. Variations of these protocols have also enjoyed successful implementation. Early work in this field was performed by Kleinrock, Tobagi, Lam and Molle (see [Klein-75a], [Klein-75b], [Klein-75c], [Lam-75], [Tobag-80]), and still is the foundation to progress made with these protocols.

2.6.1 Key assumptions

Before the discussion of the channel allocation methods, it should be mentioned that the fundamentals of the work done on this field rely on the following five key assumptions:

1. Station model: The model consists of N independent stations (computer, telephones, laptops, etc.), each generating packets for transmission. The probability of a packet being generated in a time interval Δt is $\lambda \cdot \Delta t$, where λ is the constant packet arrival rate of packets.
2. Single channel assumption: A single channel is available, and all the nodes are capable of transmitting and receiving data on it.

3. Collision assumption: If two frames are transmitted simultaneously, they overlap in time and result in a useless packet, which cannot be decoded and understood. This is referred to as a collision. All stations can detect collisions and the collided packets can be retransmitted later.

4. The time evolution of the channel can be one of the following:

   a) Continuous time: Packet transmission can take place at any time, there is no clock subdividing time into subintervals.

   b) Slotted time: Time is divided into discrete intervals (slots). Packet transmissions may take place only at the start of a slot.

5. There are only two possibilities regarding carrier sense which are:

   a) Carrier Sense: Stations can listen to the channel so that no transmission attempt is made if the channel is "sensed" as busy. Transmission will only take place if the channel is sensed as idle.

   b) No Carrier Sense: Stations cannot sense the channel before a transmission attempt is made.

2.6.2 ALOHA

ALOHA was the pioneering medium access control scheme and revealed the need for more modern schemes such as CSMA/CD and Ethernet. A station having access to a commonly shared channel transmits data effective immediately on completion of construction of a packet. If the node does not receive an Acknowledgement (ACK) within a specified period of time after transmission, a packet retransmission occurs. This method is suited for low traffic levels, but throughput deteriorates rapidly under heavier loads. Another major advantage is the ease of implementation.

Slotted ALOHA

In a slotted system the time base is divided into discrete time slots. Nodes are only capable of transmitting at the beginning of a time slot, and accordingly collisions are reduced and the throughput is increased. As there is no carrier sense, this method still proved to be inefficient. Slotted Aloha has similar advantages and disadvantages to ALOHA, and in addition the increased difficulty of implementation because of the synchronisation required.

2.6.3 CSMA (Carrier Sense Multiple Access)

This method uses "Carrier sense" to listen to whether the channel is busy or not. The station waits for the transmission to finish and, depending on the type of CSMA that

is used, then attempts to transmit at a later time when the channel is not sensed as busy.

1-persistent CSMA

When the sender is ready to transmit data, it senses wheter the physical medium is busy. If so, it senses the medium continually until it becomes idle, and then it transmits a data packet immediately.

p-persistent CSMA

This protocol is a variation of 1-persistent CSMA. When the transmitter has data to send, it continually checks whether the medium is busy. If the medium becomes idle, the sender transmits a packet, with probability p. This process is repeated until the frame is sent or another sender starts transmitting.

Nonpersistent CSMA

When the sender is ready to transmit data, it checks whether the medium is busy. If so, it backoffs (waits) for a random length of time and checks again. When the medium becomes idle, the sender starts the transmission. If a collision occurs, the sender waits a random length of time, and again checks the medium, repeating the process.

Slotted CSMA

This technique is similar to that of slotted ALOHA in that the time base is divided into fixed slots. Transmission can take place only at the beginning of a time slot and this reduces the probability of collisions, which increases the throughput. This method is rarely implemented due to the complexity of obtaining synchronisation. Results and analysis of this technique are available in [Klein-75c], [Tobag-80b] et al.

Optimization techniques

Many techniques have been suggested to optimise CSMA in order to prevent network instability for both slotted and unslotted non-persistent CSMA, by dynamically adjusting the backoff strategy. The proposed strategies all assumed that all nodes knew the status of all other nodes on the network, which is seldom the case, rendering these approaches impractical.

Another adaptation was to reject input data packets under heavy loading. In most cases this proves to be impractical.

In general, there is no practical case where all nodes possess complete knowledge of the network state. The only statistics available to the nodes is the data obtained from passing data at each individual station. Two backoff control strategies suggested to harness the network state knowledge are, firstly, the estimation of system states by means of monitoring the channel history for a given period of packet transmissions and, secondly, increasing the backoff linearly by means of any increase in observed channel traffic. Simulations have shown both these methods to be effective.

2.6.4 CSMA/CD (Carrier-Sense Multiple Access with Collision Detection)

This protocol is similar to 1-persistent CSMA. The Collision Detection added to normal CSMA implies that the station is monitoring the channel by terminating a transmission as soon as a collision is detected, and reducing the probability of a second collision on retry. The major improvement in performance resulting from use of this protocol can be attributed to nodes not having to wait for an ACK, as each node knows when a collision occurred.

Unfortunately this method requires a full-duplex channel and in many wireless systems this is impractical, for example, where there are directional antennas with narrow beam widths, that result in transmissions being hidden from other nodes on the channel. Another example is when the physical location of nodes results in "hidden terminals", where nodes are unable to detect transmission from other nodes. Collisions still occur.

2.6.5 CSMA/CA (Carrier-Sense Multiple Access with Collision Avoidance)

This protocol builds on CSMA/CD by taking further precautions to avoid packet collisions. If a node has a packet to send it scans the channel and, in the event of the channel being unoccupied, a short Request To Send (RTS) packet is transmitted to the intended receiver. If the intended receiver is not busy and the RTS is successfully received, it replies to the sender with a Clear To Send (CTS) packet. The station then responds with the data packet. If the ACK is not received within a specified period of time, the process is repeated.

Other nodes receiving the CTS are aware that the channel will be occupied with a transmission, as a result there are no other transmission attempts, and in this way collisions are reduced. This process is also known as Virtual Carrier Sense (VCS).

The fundamental principle is that RTS and CTS packets are short in comparison to the data packets. Collisions between these shorter packets are less detrimental to the system performance, as both the time the channel is occupied by collisions and

the number of collisions are reduced.

This method works for hidden terminals, but at the cost of additional overhead.

2.7 Comparison of various ALOHA and CSMA protocols

In figures 2.1 to 2.3 the parameter *a* is the ratio of the propagation time to the packet transmission time.

Figure 2.1 illustrates the decrease in throughput for 1-persistent CSMA as the propagation time increases. As *a* increases, the vulnerable period where nodes transmit simultaneously increases, and that results in an increased collision probability and reduced throughput.

Figure 2.2 illustrates the throughput for nonpersistant CSMA and is clearly superior when compared to 1-persistant CSMA. Nonpersistant CSMA reduces the competition for the channel, resulting in fewer collisions, and also increases the effective throughput.

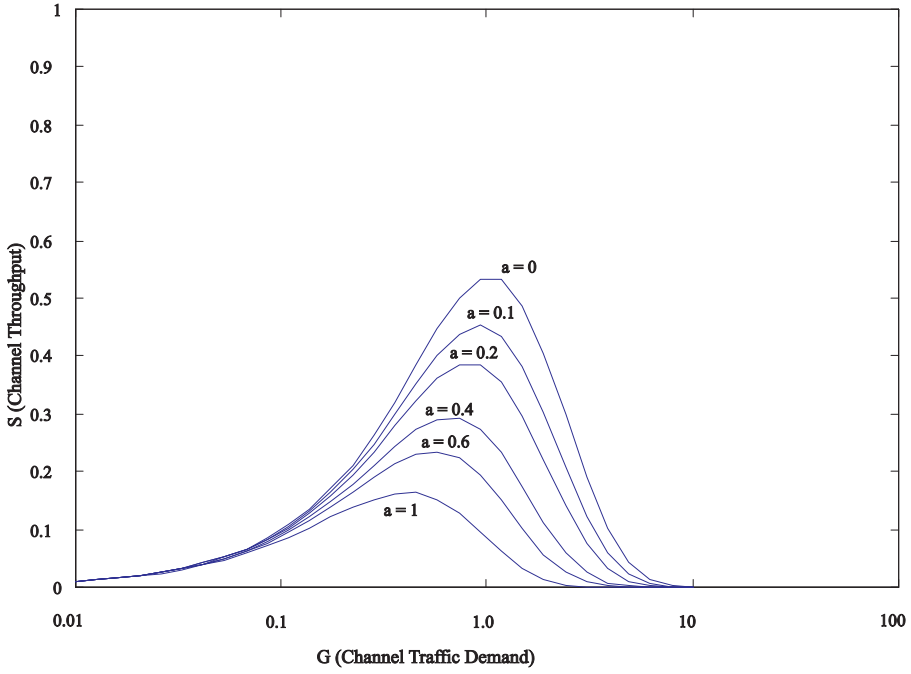


Figure 2.1: 1-Persistent CSMA channel throughput

Figure 2.3 is a comparison of various random access modes. Clearly CSMA outperforms any form of ALOHA.

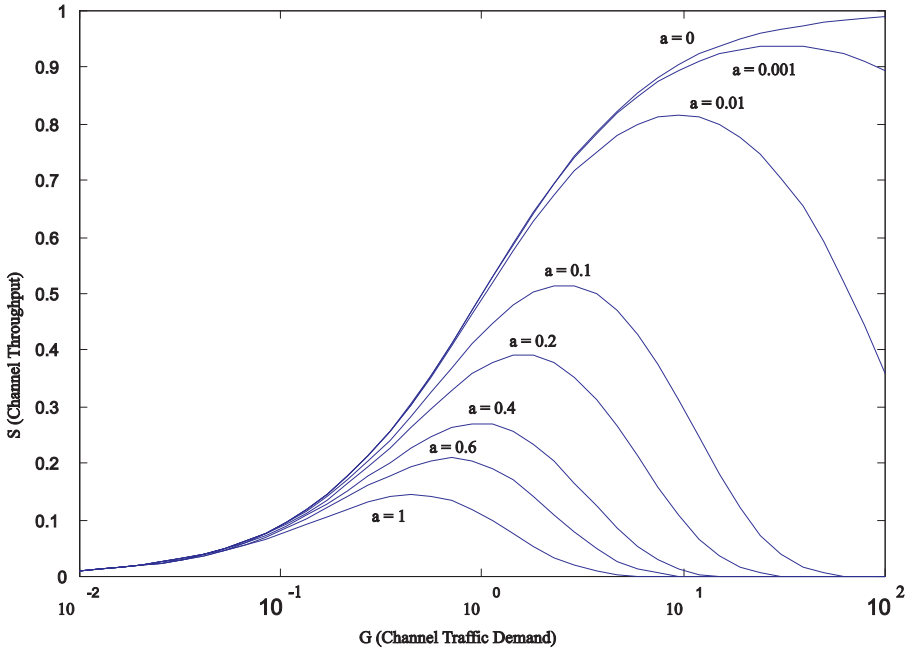


Figure 2.2: NonPersistent CSMA channel throughput

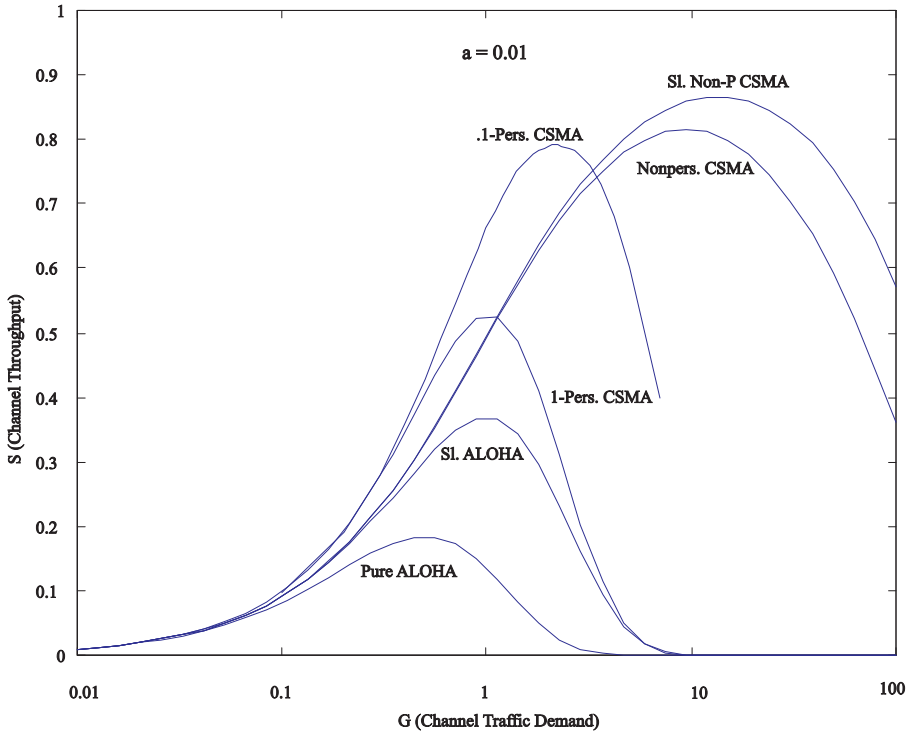


Figure 2.3: Throughput for various Random Access Modes

2.8 Typical factors influencing the selection of protocol

Many factors influence the selection of a suitable protocol. Some of these will be quantified here, with application to wireless communications.

2.8.1 Presence of Hidden Terminals

Please refer to section 3.4 for a detailed discussion.

2.8.2 Propagation delay

It was shown in section 2.7 that for various protocols the propagation delay has a significant effect on the throughput. However, for wireless communication the propagation delay is generally relatively small.

2.8.3 Channel traffic demand

The channel traffic demand has a significant effect on how certain protocols operate. At lower traffic demand rates CSMA will prove to be more effective than polling, due to less overhead being necessary than for polls, whereas polling tends to be more effective at higher channel traffic demand rates, as CSMA throughput will suffer from more packet collisions.

2.8.4 Expected number of users

The expected number of users accessing the channel simultaneously will also affect the performance of each protocol, especially in conjunction with the channel traffic demand. More nodes operating with a polling scheme at lower channel traffic demand rates will significantly decrease the throughput, with an increase of packet delay, for example when using RRP.

2.9 Summary

This section has provided a basic overview of deterministic protocols and especially RRP, which is applicable later. The various kinds of CSMA were also discussed, as this will later prove useful as a background for the IEEE 802.11 standard. The various random access protocols are also compared to each other.

Chapter 3

IEEE 802.11 background

In this chapter the necessary background will be given to the standards, terminology and physical workings of the IEEE 802.11 standard.

3.1 IEEE 802.11 standards

Table 3.1: Summary of IEEE 802.11 standards

	802.11a	802.11b	802.11g	802.11n
Standard approved by IEEE	January 2000	December 1999	June 2003	Expected 2007
Maximum data rate	54 Mbps	11 Mbps	54 Mbps	300 Mbps
Effective data throughput rate	23 Mbps	4.3 Mbps	23 Mbps	74 Mbps
Different data rate configurations	8	4	12	
RF frequency band	5 GHz	2.4 GHz	2.4 GHz	2.4 and 5 GHz
Modulation technologies	OFDM	DSSS, CCK	DSSS,CCK, OFDM	DSSS,CCK, OFDM+
Channel bandwidth	20 MHz	20 MHz	20 MHz	20 or 40 MHz
Number of channels	23	3	3	26
Number of spatial streams and antennas	1	1	1	up to 4
Typical indoor range	35m	38m	38m	70m
Typical outdoor range	120m	140m	140m	250m

Table 3.1 provides an overview of the different IEEE 802.11 standards. Some typical and nominal values are indicated. What should be noted is that these are typical values for the physical layer (OSI model, the first layer). Of importance for this project is that the IEEE 802.11b standard was chosen to work with, and from the table it becomes evident that this standard works in the 2.4 GHz range, and has a maximum data rate of 11 Mbps. The reason why this standard was chosen was that it is a proven and well implemented technology; also that the network simulator software package (*ns*2) that was used, is already well suited for this technology. It should be mentioned that all these standards use the same MAC layer protocol, which is basically CSMA, and the only minor difference is that of the n standard, which has added Quality of Service (QoS) parameters to the MAC.

3.2 General network nomenclature

This section introduces the network components and their nomenclature [see Gast-02].

Basic Service Set (BSS)

Stations are logically grouped into Basic Service Sets. When no Access Point (AP) is present, the network is a loose ad-hoc configuration called an Independent BSS. APs allow more structure by connecting disparate BSSs into a further logical grouping called an Extended Service Set (ESS).

Access Point (AP)

This is a bridge between nodes and a distribution system, where the distribution system generally refers to a wired backbone network. Bridging is the predominant function the AP provides, which converts packets from a wireless to a wired format. The AP can also provide contention free communication which is centrally controlled, and this will be discussed later.

Independent Basic Service Set (IBSS)

In an IBSS nodes connect directly with each other and do not connect to any AP. The nodes must be in direct communication range.

Extended Service Set (ESS)

Multiple APs can be joined together by using an ESS. An ESS is created by linking BSSs.

Node or Station

This refers to a fixed or mobile node with a wireless card and antenna that broadcasts data to another node/AP and receives data from it. The mobile or fixed device may be a desktop computer, laptop or mobile phone that has wireless capabilities.

Infrastructure mode

Infrastructure networks are distinguished by the use of an AP. In most IEEE 802.11 standards, if two nodes are operating in the same BSS and want to communicate with each other, all communication has to take place through the AP. If a node wants to communicate with another node in infrastructure mode it has to take two hops, that is, it first has to broadcast the packet to the AP, and then the AP will transmit the packet to the second node. Although the multihop transmission consumes valuable bandwidth, it has three major advantages.

1. An infrastructure BSS is defined by the distance from the AP. Nodes must be within range of the AP, but there is no restriction on the distance between mobile nodes. This can result in increased transmission range.

2. Allowing nodes to communicate directly with each other in the same service set would save transmission capacity, but at the cost of increased physical layer complexity, as neighbour relations will have to be maintained with all nodes. This holds true especially if a significant number of nodes are present.

3. APs can aid nodes which attempt to conserve power. Nodes using power save mode can inform the AP, whereupon the AP will buffer packets and transmit them to the node when it is active.

In order for nodes to be capable of communicating with an AP they must associate with the AP.

Ad-hoc network

When no access point is present, the network is generally referred to as loose or an ad-hoc network. Typically these networks are set up with a small number of nodes, for a short period of time.

Unicast transmission

This refers to node to node transmission. There is no transmission from a single node to multiple nodes.

Multicast transmission

This refers to the transmission of a single packet from a single node to multiple nodes.

3.3 RF Link quality

As all wireless communication is strongly dependant on the RF link quality, the IEEE 802.11 standard compensates for this, unlike most other link layer protocols, by positive acknowledgments. All transmitted frames must be acknowledged, if any part of the transmission fails, the packet is considered lost.

3.4 Hidden nodes

It is often difficult to determine the boundaries of wireless networks and a situation can occur where not all nodes can communicate with each other in the network.

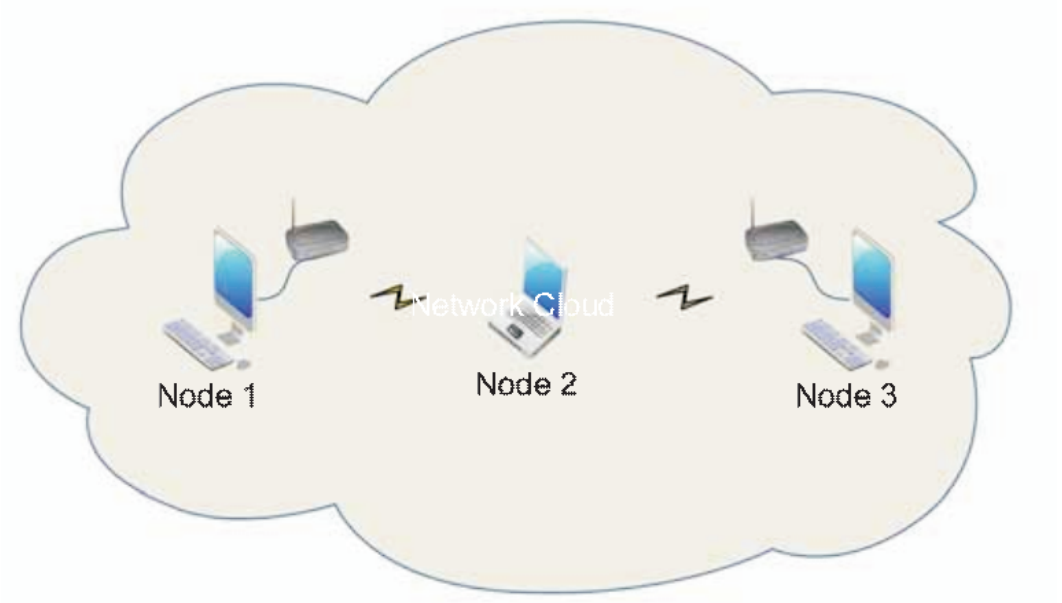


Figure 3.1: Illustration of the hidden node problem

Figure 3.1 illustrates the hidden node terminal problem, where both node 1 and 3 can communicate with node 2, but not with each other. It is easy in a network configuration such as this for nodes 1 and 3 to transmit simultaneously, causing node 2 to be incapable of making sense of the received data. Collisions in wireless network may be difficult to detect, because generally transceivers are half-duplex. To prevent collisions, 802.11 uses Request to Send (RTS) and Clear to Send (CTS) frames

(CSMA/CA).

For example, if node 1 has a packet to transmit to node 2, it first sends a RTS, node 2 replies with a CTS which notifies node 2 it can transmit and node 3 knows the channel will be occupied by a transmission for a specific period of time.

The multiframe RTS/CTS mechanism introduces a fair amount of overhead and latency, therefore it is only used in high-capacity environments and environments with significant contention on transmission. It is unnecessary for low-capacity environments.

Packets exceeding the RTS threshold are sent with RTS/CTS mechanism. The RTS threshold is a parameter set in the device driver of 802.11.

As the method outlined here was used for this project, this brief discussion should prove sufficient.

3.5 MAC Access modes and timing

Access to the channel is controlled by coordination functions. Random access (Ethernet-like) CSMA/CA access is provided by the distributed coordination function (DCF). If contention free access is required, it can be provided by the point coordination function (PCF), which is a method built upon DCF. Contention free access modes are only possible for infrastructure networks [Gast-02].

3.5.1 Virtual carrier sensing functions and the network allocation vector

In 802.11 carrier sensing is used to determine if the channel is busy. There are two types of carrier sensing, which report to the higher layers if the channel is busy. The sensing functions are physical and virtual.

Physical carrier sensing is provided by the RF components in the physical layer, where the method used depends on the type of modulation and medium. Cost restrictions make it difficult to build transceivers that can simultaneously transmit and receive data. Another impairment is the problem of hidden nodes, which makes this method less effective.

Virtual carrier sensing is provided by the network allocation vector (NAV). Most 802.11 frames contain a duration field, which indicates the period of time the channel will be busy. Stations receiving packets with a duration field set their NAV to this value. Stations set their NAV according to the time within which they expect the current operation to complete, including the time of any additional packets (ACKs,

etc.). When the NAVs value is non zero, the virtual carrier sensing function indicates that the channel is busy, and if the value is zero, the channel will be indicated as idle. An example of where the NAV proves to be useful is when a CTS frame is received by a station which did not transmit the RTS; by setting their NAV they will avoid interrupting the coming transmission.

3.5.2 Slot time

A slot time, often referred to as an idle slot time, is defined as

$$SlotTime = aCCATime + aRxTxTurnaroundTime + aAirPropogationTime + aMACProcessingDelay \quad (3.5.1)$$

A slot time has to compensate for the time taken for the receiver to do a clear channel assessment (*aCCATime*), the receiver to transmitter turnaround time (*RxTxTurnaroundTime*), the signals propagation delay (*aAirPropagationTime*), and delay caused by the time taken by the MAC to complete processing (*aMACProcessingDelay*).

3.5.3 Interframe spacing

802.11 specifies four different interframe spacings, of which three are used for medium access. Interframe spacing creates priority levels for different types of packets. Higher priority packets use shorter interframe spacing to access the channel before lower priority packets can do so. To assist with interoperability the interframe spacing is a fixed length of time and does not vary with different transmission rates.

Short interframe space (SIFS)

The SIFS is used for the highest priority frames, such as RTS/CTS and ACKs. High priority transmissions can take place after the SIFS expires. The SIFS is determined as follows

$$SIFS = aRxRFDelay + aRxPLCPDelay + aMACProcessingDelay + aRxTxTurnaroundTime \quad (3.5.2)$$

The SIFS has to take into consideration the RF propagation delay (*aRxRFDelay*), the delay that is caused by the PLCP preamble and header synchronising the receiver to the signal (*aRxPLCPDelay*), the time taken by the MAC to process the signal (aMACProcessingDelay) and the time it takes the station to turn from acting as a receiver to a transmitter (*aRxTxTurnaroundTime*)

PCF interframe space (PIFS)

The PIFS is used by the PCF mode. Stations operating in PCF mode with data can start their transmission after the PIFS time period expires. The PIFS aids the PCF in transmitting packets before contention nodes can do so. The PIFS is defined as follows

$$PIFS = aSIFSTime + aSlotTime \tag{3.5.3}$$

DCF interframe space (DIFS)

The DIFS is defined as the minimum idle channel time necessary to pass before random access nodes can access the channel. It is determined as

$$DIFS = SIFSTime + 2 \times SlotTime \tag{3.5.4}$$

Extended interframe space (EIFS)

The extended interframe space does not always have a fixed interval and is used when there is an error in a packet transmission.

3.6 802.11 basic rules

The 802.11 standard specifies a basic set of rules which is always used, and additional rules may be applied depending on the circumstances. Some basic rules are [Gast-02]

1. If the channel is sensed idle for longer than a DIFS for DCF and PIFS for PCF, transmission can take place immediately.
2. If a packet is received without any errors, the medium should be free for at least a DIFS for DCF and PIFS for PCF, before a transmission can take place.
3. Every data frame that is transmitted must be acknowledged.
4. For a transmission to be labelled a success, the complete sequence of events must be successful. For example, the sequence of event of a RTS, CTS, data and ACK all have to be without any errors (for PCF it will be a poll, data and ACK). If this is not the case, the whole process will be repeated.
5. Once a station has successfully sent the first packet of a sequence, it gains control of the channel.

6. The sending node is responsible for detecting errors and retransmissions.

DCF specific rules

1. Carrier sensing is done by both virtual and physical methods.
2. If a transmission has taken place with errors, the channel has to be idle for an EIFS.
3. If the medium is sensed as busy, the station has to wait for the channel to become idle.
4. Any packet failure increments a retry counter.
5. Any packet larger than the RTS threshold must have RTS/CTS exchange.
6. Any packets larger than the fragmentation threshold must be fragmented.
7. The retry counter is set to zero only when a CTS response is successfully received for an RTS, a positive MAC acknowledgement is received and after multicast frames.

3.7 DCF

The method by which 802.11 provides reliability is the retransmission of data packets that previously failed.

3.7.1 Retry counter

The retry counter is a method of counting the number of failed transmission attempts. The retry counter starts with an initial value of zero and with every failed transmission attempt it is increased by one. If the maximum retry value is reached, the packet is discarded, and the loss is reported to the higher-layer protocols.

Two specific retry counter limits exist, namely the short and long limits which are used individually for short or long data packets. The short limit is used with normal DCF and the long works in conjunction with the RTS/CTS mechanism.

3.7.2 Backoff counter

DCF uses an exponential backoff counter to deal with failed transmission attempts. After a transmission and a DIFS has elapsed, a contention window or backoff window follows. The backoff window is subdivided into slots, where the duration of a slot time is determined by equation 3.5.1 (note higher speed physical layers will have shorter slot times). Stations pick a random slot and wait for the slot before attempting to transmit a packet. If multiple nodes are attempting to access the

channel, the node that picks the lowest slot number will be first to transmit. The backoff window or slot is randomly chosen as

$$BC = int(rnd() \times CW(k)) \tag{3.7.1}$$

where function $rnd()$ returns a pseudo random value uniformly distributed in the range $[0,1]$ (all slots are equally likely to be chosen) and $CW(k)$ represents the contention window after k unsuccessful transmission attempts. Note that in a case where the value of the backoff counters is not an integer, the value is rounded to the next higher integer value.

The first attempt at transmitting a given packet is performed assuming a Contention Window (CW) value equal to the minimum specified value of CW_{min}. For each unsuccessful attempt, the value of CW is doubled until it reaches the upper limit of CW_{max} specified by the protocol. At the end of k unsuccessful attemps, $CW(k)$ is given by

$$CW(k) = min(CW_{max}, 2^k \cdot (CW_{min} - 1)) \tag{3.7.2}$$

Allowing long contention windows aids in keeping the protocol stable. The contention window and retry counter is reset to its minimum value only after a successful transmission or when the maximum retry count is reached where the packet is discarded.

3.8 PCF

If contention free access is required, the IEEE 802.11 standard provides a second coordination function, the Point Coordination Function (PCF). The PCF is an optional part of the 802.11 specifications; products are not required to implement it. However, the standard is designed to allow interoperation of DCF nodes with PCF.

In PCF mode the AP acts as a centrally controlled entity (polling master). In some ways, PCF resembles token based networking, with the point coordinator's polling replacing the token.

3.8.1 PCF operation

Figure 3.2 illustrates how, when the PCF mode is used, the time on the channel is divided into a Contention Free Period (CFP) and the Contention Period (CP). The CFP is used, for as the name suggests, for contention free communication, and the CP used for DCF operation. The contention period must be at least the length

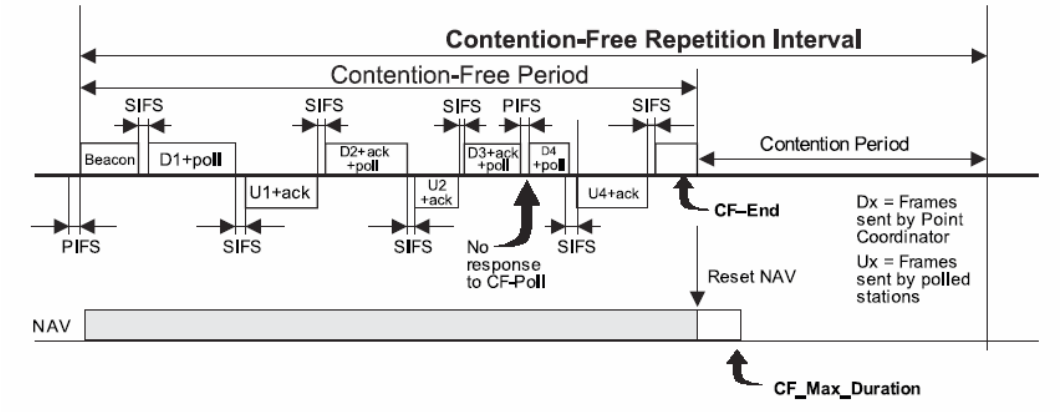


Figure 3.2: Time evolution of a superframe structure containing a CFP and CP

of the maximum-size data packet and its associated ACK. The alternating period between the CFP and CP is referred to as the contention-free repetition interval, or the superframe.

3.8.2 PCF parameters

All PCF parameters are specified in Time Units (TU), which are generally 1024 μs [Gast-02].

CFP Max Duration

This value is transmitted with the beacon, and is used to set the NAV to busy for the maximum specified duration of the CFP. This is another attempt to ensure no other unwanted transmissions take place during the CFP.

CFP DurRemaining

This is the value of the number of TUs remaining in the current superframe. Nodes use it to update their NAVs during the CFP.

CFP Count

This is a counter of how many Delivery Time Indication Maps (DTIM) will be transmitted before the start of the next CFP. Zero indicates the start of a CFP (see section 3.8.6).

CFP Period

This indicates the number of DTIM intervals between CFPs .

MinimumDataTransferTime

This is used to check after every transmission in the CFP, wheter there is enough time left to transmit another packet, or whether a CFEnd should be transmitted.

3.8.3 Polling List

After the AP takes control of the channel, it polls only associated stations on the polling list. During a CFP, stations may transmit only if the AP solicits it by transmitting a poll. Contention-free polling frames are generally abbreviated as CFPoll.

The polling list is a list of stations to be polled; stations are added to the list when they associate with the AP. Association frames contain information on the capability of the node to operate in PCF mode.

3.8.4 Typical PCF operation

In the diagram in Figure 3.2, the mechanism for contention free meduim access is shown, which starts with the AP sensing the medium. If idle, the AP waits a further PIFS period, then the AP transmits a Beacon frame which contains information such as the CFPMaxDuration which is measured in Time Units (TU) and the CFP-Duration remaining which is also measured in TU. These duration fields are used by the stations to set their NAV timers. All CFPs are started with a beacon and ended with a CFEnd packet.

After each packet transmission the AP waits a further SIFS and, if there is only enough time to send one more data packet (MinimumDataTransferTime) or less left, a CFEnd frame is sent and this signifies the end of the CFP. If there is more time left in the CFP than the MinimumDataTransferTime, then the AP goes through its polling list and when a station is determined to be next in the list, the AP checks the buffer. If data is found for that station, the AP sends a $DATA + CFPoll$ frame. If no data is found, the AP polls the station with a $CFPoll$ frame, after which the polled station is allowed a $SIFS + TimeToTxDataFrame$ (Time To Transmit Data Frame) interval to respond under normal conditions. If, after a further PIFS interval, there is no response the AP takes control and moves on to the next station in the polling list.

If a station wishes to respond to one of the polls mentioned above, and has data to send in response, it will respond with a $Data + CFAck$ frame; if it has no data to send it responds with a $CFAck$ frame. The AP, on reception of this information will, if time allows, poll the next station in the polling list. If there is data for the station, it will send a $Data + CFAck + CFPoll$ frame; if there is no data a $CFAck + CFPoll$ frame will be sent and this procedure will continue until there is just the Minimum-DataTransferTime left; on which a $CFEnd + CFAck$ frame will be sent if the AP has

received a data frame from a station. Otherwise, it can send a $CFEnd$ frame to announce the end of the CFP and all NAVs will be set to null. It is also possible for the AP node to end the CFP at any time by the use of the $CFEnd$ or $CFEnd + CFAck$ (which ever is relevant).

3.8.5 Frames used in CFP

Time in the CFP is precious and to reduce transmission overhead, different packets can be combined to form a single packet with multiple purposes, as summarised in Table 3.2.

3.8.6 TIM

The IEEE 802.11 standard uses what is called a Traffic Indication Map (TIM) to communicate to nodes that are using a power save mode to listen at specified periods to determine whether there is data available for them. These nodes in power save mode save power by not continuously listening to the channel, but listen only at specified times communicated to them by the AP. This mechanism is not applicable to this project (see [[Gast-02]]).

3.9 Packet fragmentation and reassembly

The IEEE 802.11 standard specifies methods for packet fragmentation and reassembly that can result in improved performance in noisy environments; however, this is not applicable to this project (see [Gast-02]).

3.10 802.11 frame formats

The following serves as a brief introduction to the different frame formats in 802.11. This section is to prove that the correct packet size information was used.

3.10.1 General frame structure

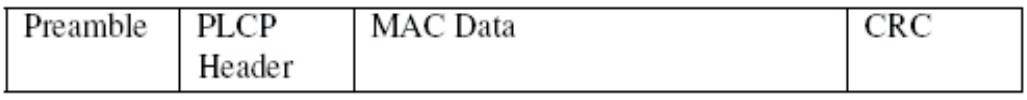


Figure 3.3: General data packet structure

Figure 3.3 illustrates the general frame structure of an IEEE 802.11 packet; the preamble is to aid timing and synchronisation between the sender and receiver. The

Table 3.2: Frames used in CFP

Frame types	Purpose
Beacon Frames	Used by the AP during the CFP to announce the AP address along with information for setting the NAVs of the stations in range.
CONTROL FRAMES	
CF-End frame	used by the AP to announce the end of the CFP
CF-End+CF-ACK frame	used by the AP to announce the end of the CFP and to piggyback an ACK to the previously received frame
DATA FRAMES	
Data	used when a station is polled, for station-to-station communication.
ACK	used to acknowledge a data transmission when station-to-station communication has occurred
Data+CF-ACK frame	used by station to send a data frame and to piggyback an ACK to the previous frame
CF-ACK frame	used by station to Acknowledge a previous frame
CF-Poll frame	used by AP to poll a station in the polling list
Data+CF-Poll frame	used by AP to poll a station in the polling list and to piggyback a data frame for that station
CF-ACK+CF-Poll frame	used by AP to poll station in the polling list and to piggyback Acknowledgement to the previous frame received
Data+CF-ACK+CF-Poll frame	used by AP to poll a station in the polling list and to piggyback a data frame for that station and to Acknowledge the previously received frame.

PLCP header contains information that is necessary for the physical layer to be able to decode the received information; it is usually transmitted at a slower data rate to ensure data integrity.

MAC Data

Frame control	Duration /ID	Address1	Address2	Address3	Sequence control	Address4	Frame body	CRC
2	2	6	6	6	2	6	0-2314	4

Figure 3.4: General data packet structure

From Figure 3.3, the MAC data frame format is shown in Figure 3.4. Note that all the fields from the figure, except the last two which are the frame body and CRC, are the MAC header of the packet. Also, the number shown with each field is the size in bytes of that specific field.

Frame control field

Protocol Version	Type	Subtype	To DS	From DS	More Frag	Retry	Pwr Mgt	More Data	WEP	Order
2	2	4	1	1	1	1	1	1	1	1

Figure 3.5: General data packet structure

Figure 3.5 is the frame control field, which is the first field of Figure 3.4. Note that the numbers in each field here represent the size in bits.

What is important in this figure is the first field, which is the protocol version that is 2 bits in size, and generally its value is zero. This field can easily be used for new types of protocol. This will be shown as applicable in a later chapter.

3.11 Summary

This section has provided an overview of the 802.11b standard which was used in the project. The basic rules and nomenclature for these types of network were discussed, as well as the working of the deterministic point coordination function and

the version of the contention protocol used, which is the distributed coordination function. Important things that were highlighted were determining when a transmission was labelled a success, error detection and carrier sensing.

It was shown in Chapter 2 that non-persistent CSMA was superior to 1-persistent CSMA; therefore, it is no surprise to see that the 802.11 standards commission adopted this method, described in this chapter, with some changes to compensate for the shortfalls of wireless.

Chapter 4

Simulation tools

From Chapter 3 it could be seen that DCF is a contention protocol, which makes mathematical modelling difficult. Further, the combination of contention nodes in conjunction with deterministic nodes makes the analysis even more difficult; therefore, it is imperative to obtain an accurate simulation model.

In this section a few different simulation tools will be discussed, as well as the construction of the simulation model.

4.1 Simulator requirements

For the identification of a simulation tool for the purposes of this project, the following requirements, as outlined here, should be met:

Suitability for modelling the system

It will be advantageous if the standard simulator comes with the necessary tools for the purposes of the project; this includes incorporating additional libraries and toolboxes. Choosing a more established simulator increases the chances of obtaining the necessary purpose built tools.

Proven track record

Most freeware simulation tools come with the possibility of incorporating a fair number of programming errors. Using a simulator that has a proven track record, and receives a fair amount of use and attention, increases the chances of obtaining accurate results and less programming errors.

Reproduction of error

It is critical for the simulator to reproduce the errors, especially, because the pro-

tocol used in this project is the IEEE 802.11 protocol, which is a random access protocol. This implies that the seed to the random number generators should be user specified, in order to be able to recreate results for scrutiny.

Configuration and scalability

The scalability and configuration of a network is of the utmost importance; the more options available, the better chances are of customising the simulation to the exact purpose.

Documentation

Concise, easily understood literature can have a significant impact on the time required for the completion of a project, therefore it is important.

Suitability to packet-switching networks

There are many discrete event simulators on the market, but there are simulators that have been specifically designed for packet-switching networks which should ease implementation.

4.2 Identified simulation tools

The two main simulators identified as suitable were *OMNeT++* and *ns2*.

OMNeT++

OMNeT++ is a public-source simulation environment, the main goal of which is the simulation of communication networks. However, its design is quite open, which enables other target applications. It has a sophisticated Graphical User Interface (GUI) and common models like IPv4, IPv6, Ethernet, MPLS etc. are available.

OMNeT modules are structured by an own network definition language NED, while functionality is coded by using *C++* classes.

ns2

ns or the network simulator (also called *ns2*, referring to the current version) is a discrete event simulator. *ns2* was primarily designed to simulate TCP/IP networks, and its general uses are for ad-hoc networks and routing. There is an array of support for network protocols, offering simulation tools for wireless and wired networks alike.

ns uses a scripting language called *Tcl* to provide scalable input parameters, while fixed parameters are specified in *C++* to improve speed.

Comparison of identified simulators

In the initial stages, *OMNeT++* was first used and proved to be a user friendly simulator. However, as the project progressed, it was found that *OMNeT++* did not have any PCF functionality and not enough literature was available to enable the creation of a PCF network, therefore *ns*2 was finally chosen. *OMNeT++* did not have IEEE 802.11 as part of the original package; the INET framework had to be installed to obtain this functionality, which proved cumbersome.

*ns*2 has been around for roughly fifteen years and comes with a wealth of literature and resources. It also has a toolbox for the simulation of a PCF network created by Anders Lindgren. Another advantage *ns*2 has over *OMNeT* is that it does physically construct the IEEE 802.11 packets and interprets them, where changes in bit values proved useful. *OMNeT++* determined only the duration of a packet transmission.

4.3 PCF toolbox by Anders Lindgren

The PCF toolbox was created in 2001 and adds PCF functionality to the original *ns*2 802.11 MAC. It was found that this toolbox was not compatible with newer versions of *ns*2 and it had to be used with the *ns*2 version it was created with (ns-2.1.b8a). This version of *ns*2 supported only the IEEE 802.11b standard and alterations had to be made, for example, the maximum data rate was 2 Mbps and had to be changed to 11 Mbps.

The toolbox provides functionality for creating and adding nodes to the polling list, creating a point coordinator, and specifying parameters such as the superframe duration and the maximum duration of the CFP.

4.4 ns2 configuration

All variable input parameters for *ns*2 were to be specified from a Tool Command Language (*Tcl*) script that hooks into the *C++* code.

4.4.1 General IEEE 802.11 node configuration

Figure 4.1 illustrates the plumbing of a general wireless node using the DSDV (Destination Sequence Distance Vector) routing protocol ([FV-08]). Each block provides a modular function and various options for operation. Also indicated, are the pointers used to interface between the modular functions. For example, if the MAC wanted to access the interface queue (IFq) to obtain the queue length, it would have

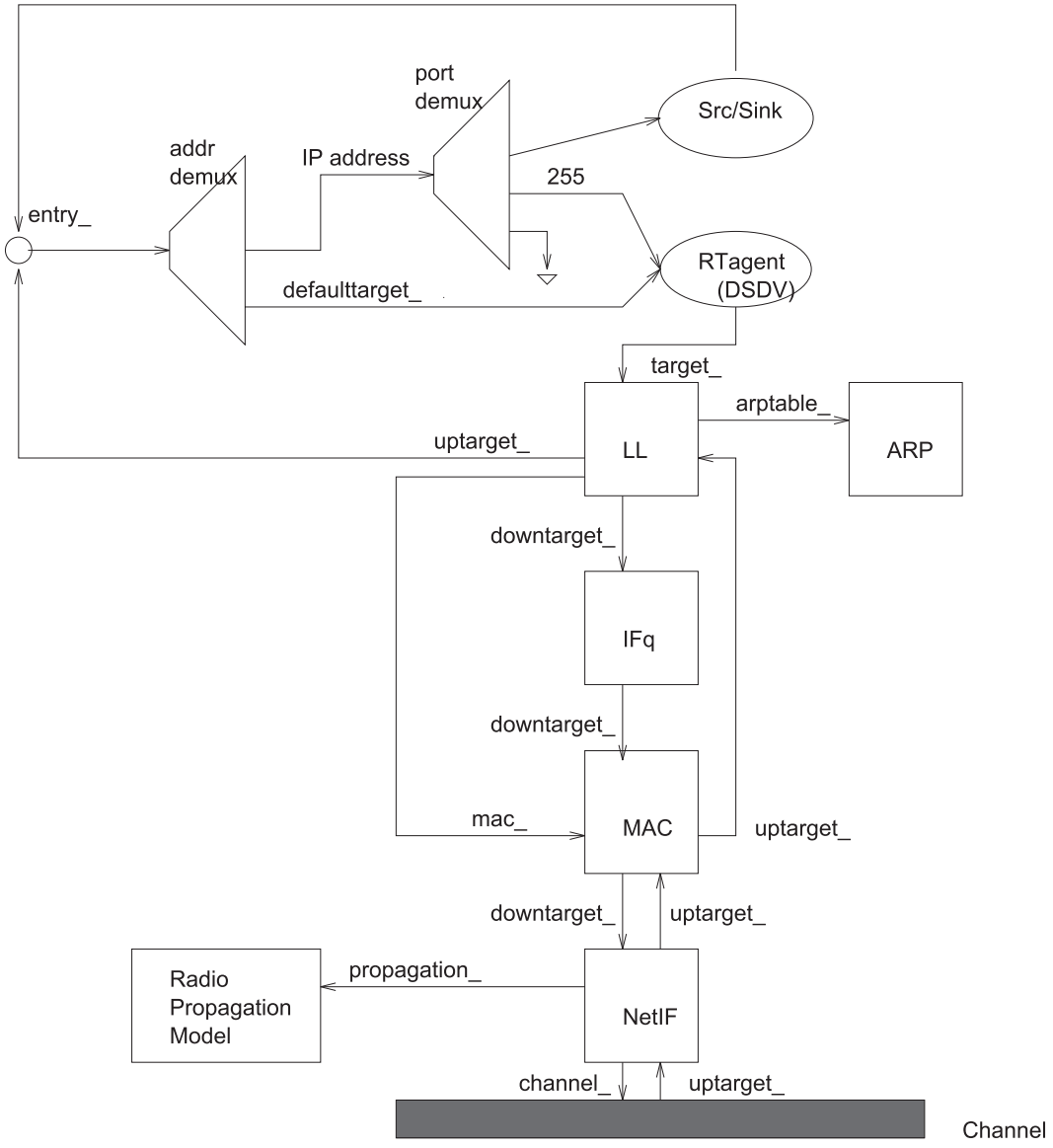


Figure 4.1: Schematic of a wireless node under CMU monarch's wireless extension to ns2

to send the request to the Link Layer (LL) through the MACs uptarget_ pointer and the LL will then request the queue length through its downtarget_.

A short description of each module will follow, with some of the possible options available, and the option chosen.

Radio propagation model The option used was simple free space loss. This provided a near perfect channel which would make mathematical modelling easier because no compensation is necessary for noise, multipath or any other losses. The command for *ns*2 in *Tcl* is: "set opt(chan) Channel/WirelessChannel".

Antenna The antenna is not shown on the schematic, but it forms part of the NetIF (Network Interface). The option used was a simple omni-directional antenna. The command: "set opt(ant) Antenna/OmniAntenna").

Network interface (NetIF) The Network Interface layer serves as a hardware interface which is used by a mobile node to access the channel. The wireless shared media interface is implemented as class Phy/WirelessPhy. This interface is subject to collisions and the radio propagation model receives packets transmitted to the channel by interfaces of other nodes. The model approximates the DSSS radio interface (LucentWaveLan direct-sequence spread-spectrum). The *Tcl* command is: "set opt(netif) Phy/WirelessPhy".

MAC The MAC layer only has the option of using the IEEE 802.11 b standard with the version of *ns*2 that was used for this project. The inclusion of the PCF toolbox still requires the same *Tcl* command, which is: "set opt(mac) Mac/802_11".

Interface Queue (IFq) The class PriQueue is implemented as a priority queue which gives priority to routing protocol packets, inserting them at the head of the queue. No routing was required for this project and a simple droptail queue was implemented in which if the number of packets in the queue exceeded a predefined value, a new packet arrival will be discarded (*Tcl* command: "set opt(ifq) Queue/DropTail").

Address Resolution Protocol (ARP) This module receives queries from the Link layer. If ARP has the hardware address for a destination, it writes it into the MAC header of the packet. Otherwise it broadcasts an ARP query, and caches the packet temporarily. For each unknown destination hardware address, there is a buffer for a single packet. In case additional packets to the same destination are sent to the ARP, the earlier buffered packet is dropped. Once the hardware address of a packet's next hop is known, the packet is inserted into the interface queue.

Link Layer (LL) The link-layer object is responsible for simulating the data link protocols. Many protocols can be implemented within this layer, such as packet

fragmentation and reassembly, and reliable link protocol. Another important function of the link layer is setting the MAC destination address in the MAC header of the packet. In the current implementation this task involves two separate issues: finding the next-hop-node's IP address (routing) and resolving this IP address into the correct MAC address (ARP). For simplicity, the default mapping between MAC and IP addresses is one-to-one, which means that IP addresses are re-used at the MAC layer. There is only one LL option in *ns*2 which is: "set opt(ll) LL".

Routing Agent (RTAgent) The version of *ns*2 used requires that a routing protocol be specified. It was decided to use the DSDV protocol. This project only uses infrastructure mode, which implies that as there will only be a single hop between nodes, any routing protocol is irrelevant (*Tcl* command: "set opt(adhocRouting) DSDV").

Source and sink (src/sink) The choice here is to use a either a TCP (Transfer Control Protocol) or UDP (User Datagram Protocol) packet source. It was decided to use UDP for its simplicity, and whichever of the above mentioned is used will have no effect on the results. The source chosen, as will be illustrated in section 5.1.3, was a Poisson traffic source to aid with Markov modelling. The source is paired with a sink which is responsible for the destruction of successfully received packets. This is also sometimes referred to as the agent.

Address and port demultiplexing (addr and port demux) This is used by *ns*2 to determine if the packet received is destined for this node, in which case it is then passed to sink or, if not, it is forwarded to the routing agent to pass it to the lower layers for transmission.

4.4.2 Statistics collection

To collect statistics from simulation, *ns*2 provides what is called trace support. As *ns*2 is a discrete event simulator, each time an event occurs this data is traced. The events that the standard *ns*2 wireless LAN provides for monitoring is tracing packets that are sent, received and dropped by agents, routers and the MAC layer. The output of the traced events is a text file with a standard format as determined by *ns*2 (see [FV-08]).

It should be mentioned that the method of tracing implemented by *ns*2 was not sufficient to collect all the statistics necessary for the purpose of this project (refer to section 4.4.6). To compensate for this, additions were made to the *C++* code of *ns*2, to enable collection of data of events by printing them to text files. These text files were then used for processing to collect the necessary statistics.

The reason why the data not provided by tracing could not be added to the trace file is that in order to trace objects they must belong to the class TracedVar. To add

the variables necessary to this class would require a major rework of the code and has no real benefit over the method that was used.

4.4.3 Processing of trace files and collected data

Two programs were identified to do processing of the collected data, namely *Perl* and *AWK*. Both programs are sufficient for the purposes of this project, but *AWK* was chosen because of easy implementation and the wealth of examples available specifically for *ns*2 applications.

*ns*2 provides a command "set tr" which processes the *ns*2 trace support data at the end of each simulation run. The format of the command is as follows

set tr [open "| gawk -f userscript.awk inputdata»outputfile" w]

From the same script, the data that was collected by the additions made to the code made for the purpose of this project is also processed with each simulation run.

4.4.4 Additional Tcl configuration parameters

Additional parameters that need to be specified from the *Tcl* script are

- The start and stop time of the simulation
- The packet size
- The positioning of the nodes and if mobility (node movement) should be included. No mobility was used for this project
- The number of wireless nodes that should be created
- Wired routing to other networks not in the same BSS. This option was not used and is not necessary

4.4.5 Additional configuration parameters in the C++ code

*ns*2 comes with a set of default values which are used to initialise all variables, otherwise the code will not compile. The following is a list of the variables that are of particular importance, which are found in the mac-802_11.h file are:

- The minimum and maximum contention window sizes (DSSS_CWmin and DSSS_CWmax).
- The slot time (DSSS_SlotTime)
- The clear channel assessment time (DSSS_CCATime)

- The receiver to transmitter turnaround time (DSSS_RxTxTurnaroundTime)
- The SIFS time (DSSS_SIFSTime)
- The packet preamble and PLCP header lengths (DSSS_PreambleLength)

4.4.6 Required statistics

The main aim of the simulation is to acquire the necessary statistics, which were identified as follows

- Throughput, which could be obtained with the *ns*2 tracing method
- The packet delay, which is the time necessary from when the packet is fetched from the head of the queue until the reception of the ACK indicating a successful transmission (*ns*2 tracing could not do this and our own methods was used to obtain the data)
- The end-to-end packet delay, which is timed from the moment the packet arrives at the queue until the reception of the ACK, indicating the successful transmission (the method had to be devised, as *ns* did not provide the necessary support)
- The average buffer occupancy for the determination of which, again, our own method was necessary.
- The collision and packet dropping probabilities, which the trace format for *ns*2 provided support for.

It should be mentioned that later versions of *ns*2 does have the necessary functions to obtain the delays and the buffer occupancy but unfortunately not this version. These functions are also available for wired networks in the version of *ns*2 that was used, but not for wireless.

4.5 Matlab

All statistics collected from *ns*2 were lastly imported into Matlab for post processing. Matlab was used to determine the error between the simulated and the mathematical results, to plot data and for mathematical manipulation of results obtained from mathematical modelling. The mathematical manipulation was required to adapt the results to model the designed network and the protocols used.

4.6 Summary

This Chapter describes the decision making process which led to the choice of the *ns*2 simulator for use in this project. A general overview of *ns*2 is also provided, that takes into consideration the necessary configuration, simulation parameters and the procedure used.

Most of the simulation parameters will be discussed in more detail in the following chapter, which will provide better clarity.

Chapter 5

Simulations

In this chapter the simulation of an IEEE 802.11 network in *ns*2 for both DCF and PCF modes will be discussed. All simulation parameters, results and conclusions will also be further described.

5.1 Experimental setup

5.1.1 General positioning of nodes

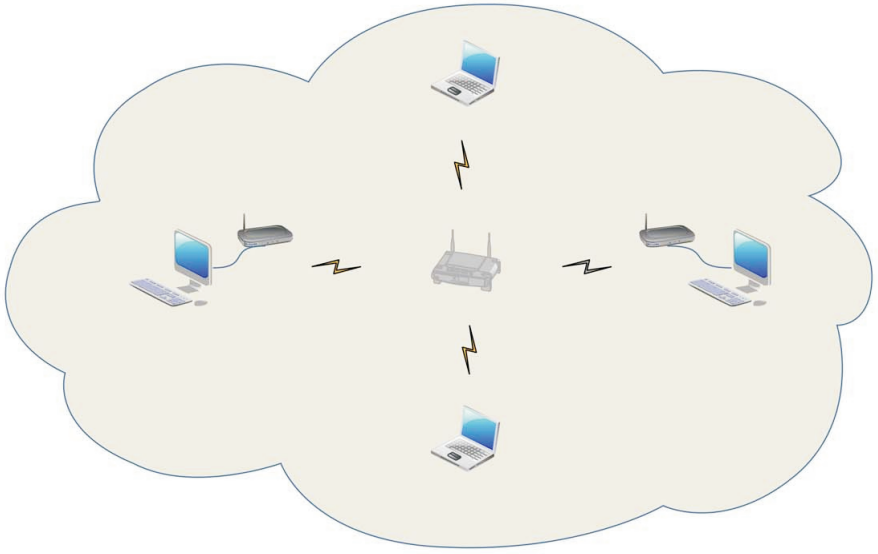


Figure 5.1: Example of a wireless network where nodes are symmetrically spaced at equal distances from an AP

The approach used for the experimental setup was to use a central Access Point (AP) as a Base Station (BS). All nodes are spaced symmetrically on a circle at 60m from the BS, which leads to a fixed propagation delay, which aids mathematical modelling. The nodes communicate only with the BS and not with one another; neither does the BS send any data packets; it sends only control frames such as beacons, polls, and ACKs.

To ensure that the BS and all other nodes are within hearing distance of one another the transmission power was set to 0.3 Watt, with omni directional antennas having a 0 dB gain.

The transmission channel is set to be error free with no noise. The only signal attenuation is due to free space loss. The loss due to connectors and cabling at the chosen transmission power is also negligible. No objects that could cause multipath were placed in any of the transmission paths and all nodes are stationary (no mobility) at all times.

5.1.2 Simulation link budget

To prove the correctness of the statements in section 5.1.1, a simple link budget is necessary. A link budget is used to determine how much stronger the received signal is than all the other noises and losses in the system. This gives an idea of how reliable the communication over any link might be. The result is expressed as a signal-to-noise ratio (SNR) and the general unit of measurement is decibels (dB).

Free-space loss (FSL)

This is generally the largest contributor to signal strength loss. It is calculated as

$$FSL = (\frac{\lambda}{4\pi d})^2 \tag{5.1.1}$$

where d is the distance over which transmission takes place, and λ the wave length.

$$\lambda = \frac{c}{f} \tag{5.1.2}$$

where c is the speed of light constant and f is the carrier frequency of the signal.

Substituting λ into FSL gives

$$L_{FSL} = (\frac{4\pi}{c})^2 (fd)^2 \tag{5.1.3}$$

The FSL equation can be rewritten in units of dB as

$$FSL = -27.55(dB) + 20 \cdot log[frequency(MHz)] + 20 \cdot log[distance(m)] \tag{5.1.4}$$

Friis free-space equation is defined as

$$Pr = \frac{P_t G_t G_r}{L_{FSL} L_{eq}} \tag{5.1.5}$$

where P_r is the power at the receiver, P_t the transmitted power, G_t the antenna gain at the transmitter, G_r the antenna gain at the receiver, L_{FSL} the free-space loss, and L_{eq} the losses due to equipment.

This equation can be rewritten in units of dB as

$$Pr = P_t + G_t + G_r - L_{FSL} - L_{eq} \qquad [dB] \tag{5.1.6}$$

A typical receiver sensitivity threshold for an IEEE 802.11b access point, as shown in Appendix A, at the maximum transmission rate of 11 Mbps, is -85 dBm. The frequency we chose to work in was at 2.4 GHz. The antenna gains and equipment losses were set to zero in *ns*2

$-85dBm = 24.5 - L_{FSL}$

$L_{FSL} = 109.5dB$

$L_{FSL} = -27.55 + 20log(2.4 \times 10^3) + 20log(d) = 109.5$

$d = 2968m = 2.968km$

The distance of 2.968 km proves that the selected transmission power is more than sufficient to guarantee that all stations are within range of one another, especially

with the furthest distance between two nodes being 120m. This will ensure there are also no hidden terminal problems.

5.1.3 Traffic source used

In order to later be able to investigate a mathematical model to emulate and predict the behaviour of a wireless network, traffic will be generated with a Poisson distribution. The exponential on/of traffic generator of *ns*2 was configured to do this. Please refer to section 6.2.3 for a detailed discussion of the reasoning behind the choice of using the Poisson distribution. The source for the simulation was also configured so that there would be no bursty traffic; that is, more than one packet cannot be generated in a single time instant.

All figures for delay, buffer occupancy and throughput are plotted against the packet arrival rate of a single node (λ) which is measured in packets per second (pkts/s) or against the combined offered load (Λ) for all nodes, which is measured in bits per second (bps). The two metrics are related by $\Lambda = \lambda \cdot N \cdot E_p$, where λ is packet arrival rate for all nodes, N is the number of nodes and E_p is the average packet size. This equation was only used if all nodes had the same packet arrival rate.

5.1.4 Statistical approach

All simulations were run for 80 seconds, to ensure enough time to collect accurate and stable results for averages of certain performance metrics.

Statistical collection only starts after 20 seconds from the start of the simulation. This gives all stations enough time to associate with the base station and to remove all transient elements that will influence average values.

The random number generators were provided with identical seeds, to ensure that results could be reproduced.

5.1.5 Buffer management

Each wireless node has a buffer to queue packets to be transmitted, which works on a First In First Out (FIFO) principle. If the number of packets queued in the buffer exceeds a predefined value the packets will automatically be discarded, which is referred to in *ns*2 as a droptail queue.

5.1.6 Routing

As was discussed in section 4.4.1, *ns*2 requires that a routing protocol be specified. However, as this is not a multi hop network, the choice is irrelevant.

5.1.7 Cause of transmission errors

The channel that will be used is outlined in section 4.4.1 and the only signal losses are free space losses. As the link budget outlined in section 5.1.2 illustrates, these losses will not be the cause of any lost packets, neither will there be hidden terminal problems. The simulation will be configured so that there will be no losses due to multipath or noise on the channel. The hardware also has negligible losses and there is no loss due to hardware imperfections such as problems with synchronisation.

The only error detection for frames is a Cyclic Redundancy Check (CRC), which does not really have any significant error correcting capabilities; accordingly, all frames received in error have to be retransmitted.

5.2 DCF Simulation Parameters

As the scope of this project did not include addressing the effects of different parameters, as outlined here, the standard values as specified in the IEEE 802.11b standard or similar were used.

Table 5.1: Chosen ns2 simulation parameters

Parameter	Value
CWmin	31
CWmax	1023
Slot Time	20 microseconds
SIFS Time	10 microseconds
Mac Short Retry Limit	7
Mac Long Retry Limit	4
Preamble Length	144 bits
PLCP Header Length	48 bits
Data Rate High	10.1 Mbps
Data Rate Low	2 Mbps
PLCP Data Rate	2 Mbps
Max buffer size	20
Propagation delay	0.2 microseconds
Receiver Transmitter Turnaround time	5 microsends

5.3 DCF Simulations

The following sections will provide the results obtained from *ns*2. These will provide the reader with a general overview, and an exact analysis with numerical results

will be given in Chapter 6 .

5.3.1 DCF throughput

5.3.1.1 Goodput and throughput definition

The definition for goodput used is the useful throughput, implying the throughput excluding all headers, dropped and retransmitted packets, and control packets such as ACKs. This is only the data that is useful for the user, excluding any overhead or wasted packets. Goodput is often referred to as the raw bit rate, and its unit of measurement is bits per second (bps).

From this section onward and through all the following chapters, goodput and throughput will be used interchangeably but what is referred to is actually the goodput. Goodput was used rather than throughput, because throughput is often not a true measurement of system performance.

5.3.1.2 Factors on which the goodput will be dependent

With the parameters set to a fixed value as described in section 5.2, the factors that will influence the goodput are the packet arrival rate, the packet size and the number of wireless nodes contending for the channel. The units of measurement that will be used for the traffic arrival rate will be either packets per second or bits per second.

The reason that only these parameters will influence the performance is that a fixed propagation delay is specified (there is no node movement), the channel is specified to be ideal and all equipment is ideal, implying that no data is lost due to errors caused by hardware failure.

5.3.1.3 DCF goodput results and discussion

As can be seen from the simulation results (see Figures 5.2 and 5.3), the goodput of the system, if the parameters explained in Table 5.1 are fixed, are influenced by the following:

Number of nodes

The saturation goodput value is slightly higher for fewer transmitting nodes. Fewer nodes transmitting reduces the probability that any two or more nodes try to transmit a packet at the same time, resulting in fewer packet collisions, which in turn increases the goodput.

Packet size

Larger packet sizes increase the saturation throughput value. Fewer packets are

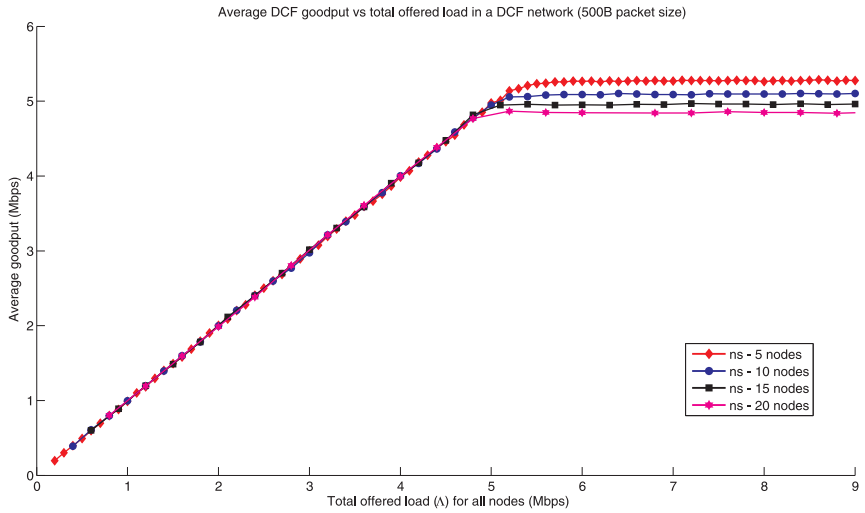


Figure 5.2: Goodput versus the total offered for all transmitting stations (packet size 500B)

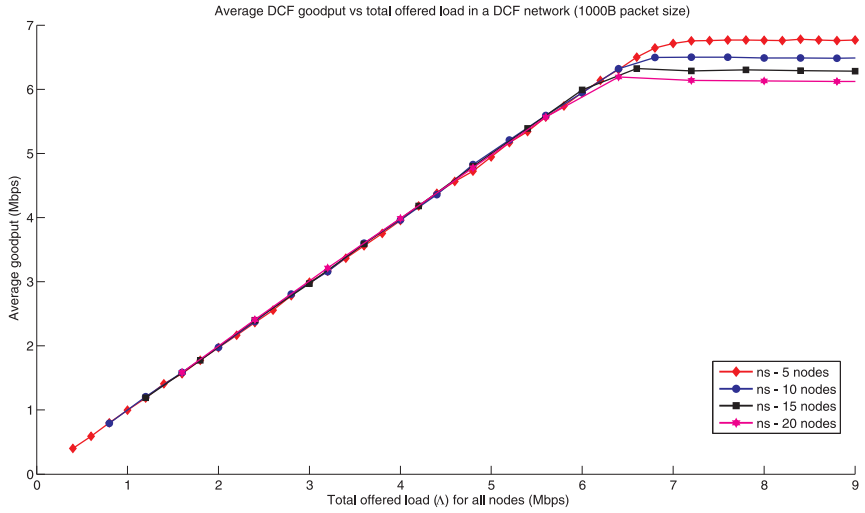


Figure 5.3: Goodput versus the total offered for all transmitting stations (packet size 1000B)

transmitted in a single second with larger packet sizes and every packet transmitted contains more data, thus decreasing the overhead that is required, such as ACKs and headers, which in turn increases the saturation goodput value.

Offered load

As was discussed in Chapter 2, the goodput follows the offered load almost linearly up to the saturation point, where the system reaches its maximum goodput value and remains roughly at this value. The results are in agreement with the non-persistent CSMA theory for goodput.

5.3.2 DCF packet delay simulation results

5.3.2.1 Packet delay definitions

The packet delay is defined as the time duration from the instant when the packet is removed (dequeued) from the packet buffer by the MAC up until the ACK is received, implying a successful packet transmission.

The end-to-end delay is defined as the time duration from the instant when the packet is removed from the packet buffer until the ACK is received, implying that the packet was successfully transmitted.

It should be mentioned that from here on for all the following chapters the buffer size has a maximum of 20 packets which can be accommodated.

5.3.2.2 Statistical approach

The statistical approach is the same as is described as in section 5.1.4. It should be mentioned that starting to log the delay results only from 20 seconds gave enough time for the packet delay values to stabilise and reach steady state, and by doing so, to remove all transient components.

5.3.2.3 Additions made to ns2 code

The version of *ns*2 that was used for this project did not have the necessary support to collect the data for the packet delay. Each data packet in *ns*2 receives a unique identification (id) number, the acknowledgement packet acknowledges the successful reception of the packet with the same number. To obtain the results, the additions to the *ns*2 code for the packet delay in the mac-802_11.cc for every node, was that, after the timer expired for the successful reception of the ACK, it was logged to a separate text file with the time and the unique packet id, and every time a packet was dequeued from the queue.h file, the same was done to a different text file. The text files were then processed by AWK to obtain the average delays.

5.3.2.4 Factors on which the DCF packet delay will be dependant

With the parameters fixed as explained in section 5.2, the parameters that will influence the packet delays are the packet arrival rate, the buffer capacity (length), the number of stations and the packet size.

5.3.2.5 Packet delay results and discussion

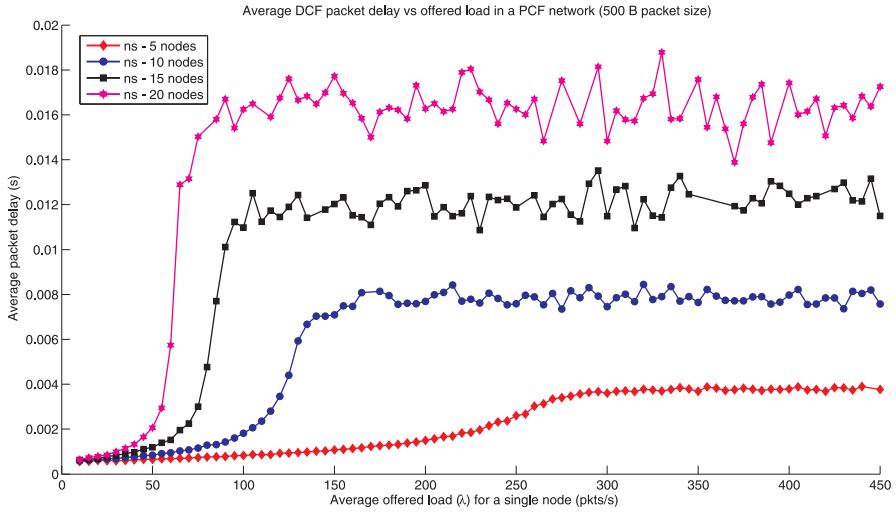


Figure 5.4: Average packet delay versus the packet arrival rate (packet size 500B)

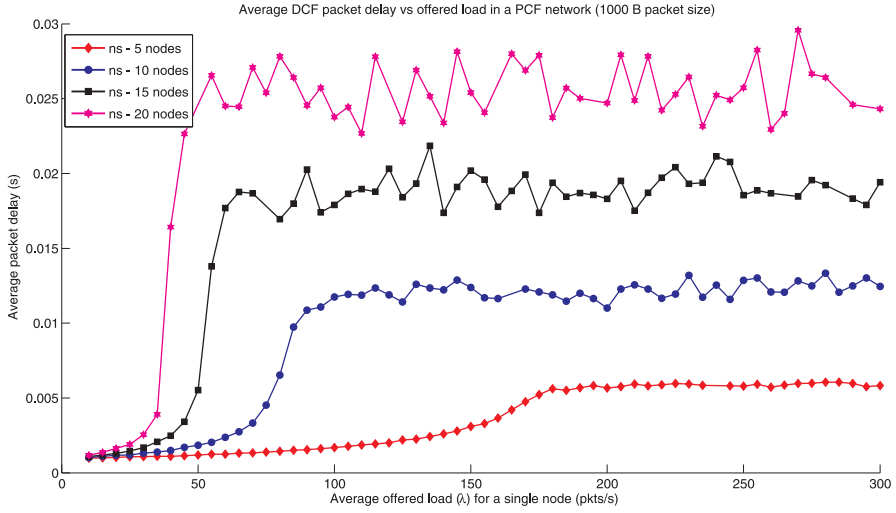


Figure 5.5: Average packet delay versus the packet arrival rate (packet size 1000B)

From the figures (see figure 5.4 and 5.5), it becomes evident that the delay of a packet is influenced by:

Number of nodes

As the number of nodes is increased, the packet delay should also increase. The results are in agreement with this. The reason for this correlation is that as the number of nodes competing for the channel increases, the larger the probability becomes for packet collisions. The larger the probability of a packet colliding becomes, accordingly, the time spent backing off increases which, results in an increased packet delay and, when the increase in packet delay starts and at the rate it increases. The relationship is, unfortunately, not linear, as this is not a Jackson network, therefore a different mathematical approach will be necessary to quantify this occurrence.

Packet size

It is evident that the larger the packet size, the larger the packet delay would be. As a fixed transmission rate was used, automatically it would take longer for a packet to be serviced which, in turn, would result in a longer queueing delay. Larger packet size implies that fewer packets are transmitted during a second of sampling which, in turn, will result in less overheads such as ACKs and headers, and less packets lost due to collisions. The implication of this is that there is not a linear relationship between size of a packet and its delay.

5.3.3 DCF buffer occupancy simulation results

5.3.3.1 Additions made to ns2 code

The version of *ns*2 that was used for this project did not have the necessary support to collect the data for the buffer occupancy. To obtain the required results, the additions to the *ns*2 code for the average buffer occupancy were made in the queue.h file. After a predefined period of time which is necessary to ensure no transient components will influence the results, the queue length is monitored and the total time spent in each buffer occupancy is logged. With the completion of the simulation the average buffer occupancy is determined and logged by AWK.

5.3.3.2 Factors on which the DCF buffer occupancy will be dependant

With the parameters fixed as explained in section 5.2, the parameters that will influence the buffer occupancy are the packet arrival rate, the buffer capacity (length), the number of stations and the packet sizes.

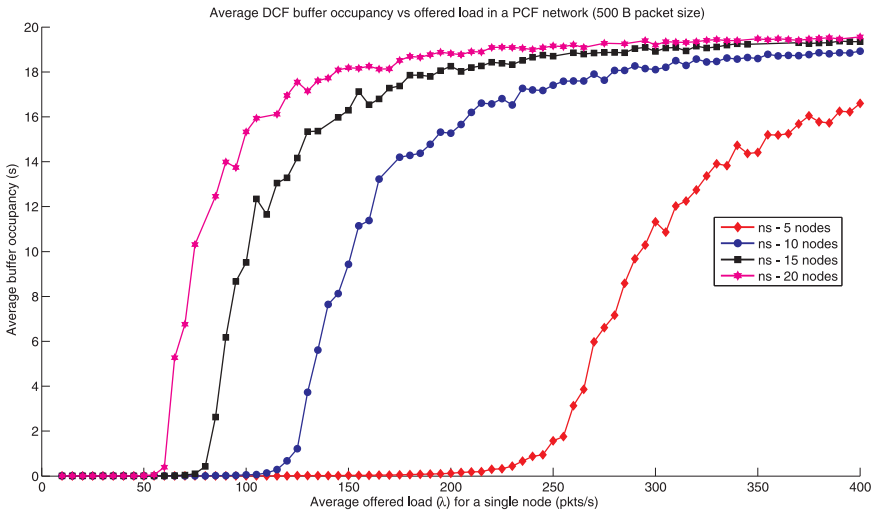


Figure 5.6: Average buffer occupancy versus the packet arrival rate (packet size 500B)

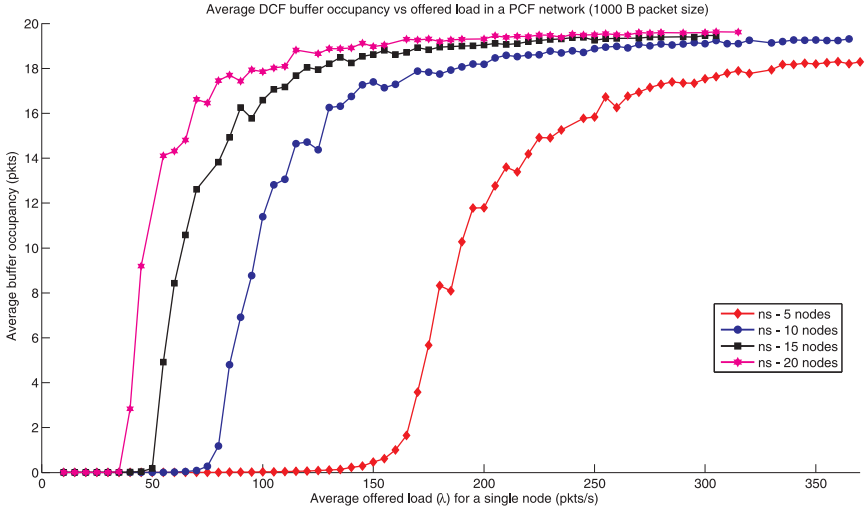


Figure 5.7: Average buffer occupancy versus the packet arrival rate (packet size 1000B)

5.3.3.3 DCF buffer occupancy results and discussion

The buffer occupancy results prove to be in agreement with the packet delay. As soon as the packet delay reaches its saturation value, the buffer occupancy starts increasing. The dependency on the number of nodes and the packet size of the average buffer occupancy increases the rate at which saturation is reached, and is also in agreement with that of the DCF packet delay. It should be noted that the buffer occupancy and packet delay are related by Little's law, as discussed in section 6.2.4. Lastly, as this is a blocking system in that if the buffer is full any new packet arrivals are blocked, the buffer occupancy will not reach a value larger than 20 packets, which the results confirm.

5.4 PCF simulations

5.4.1 Overview and changes made to the ns2 PCF toolbox

The *ns*2 PCF toolbox used is designed by Anders Lindgren. The approach he used was to create a polling list of the nodes that are to be polled by specifying this from the same Tcl script used to initialise the other simulation parameters. It was found that what the code did at the MAC layer was to allow a station on the polling list to transmit during the CFP and during the CP (see Figure 5.8). This was a problem to model mathematically and would also have a significant influence on the performance of such a network. In addition, instead of pausing the backoff timers of a station not on the polling list when a beacon was received, and after the CFEnd frame was received, resuming backoff timers from their former states, these timers were reset when the CFEnd frame was received. At low traffic rates this increased the throughput but, at higher traffic rates, the throughput was reduced.

Some changes were made to the code to ensure that pollable stations only transmitted during the CFP (see Figure 5.9). After several failed attempts, the approach used was to give stations a Boolean variable by which they could monitor whether they were pollable. The variable is set to one if the station is polled every CFP, and to zero if it does not receive a poll frame with its address. When the CFEnd frame is received, if a station is pollable and has a packet to transmit, it keeps increasing its backoff counter until the next CFP. After a successful packet transmission by a pollable station the backoff counter is again initialized back to a value between 0 and $CW_{min} - 1$. Another change made for the nodes that are not on the polling list and only operate in the CP, was that instead of stopping the backoff timers, they were paused when a CFP was started and resumed normally at the values they were at when the CP was restarted. This approach works as long as the CP period is not longer than the time it takes for the packet to go through all the backoff stages and be discarded.

By using the above approach it was found that pollable nodes did not send associa-

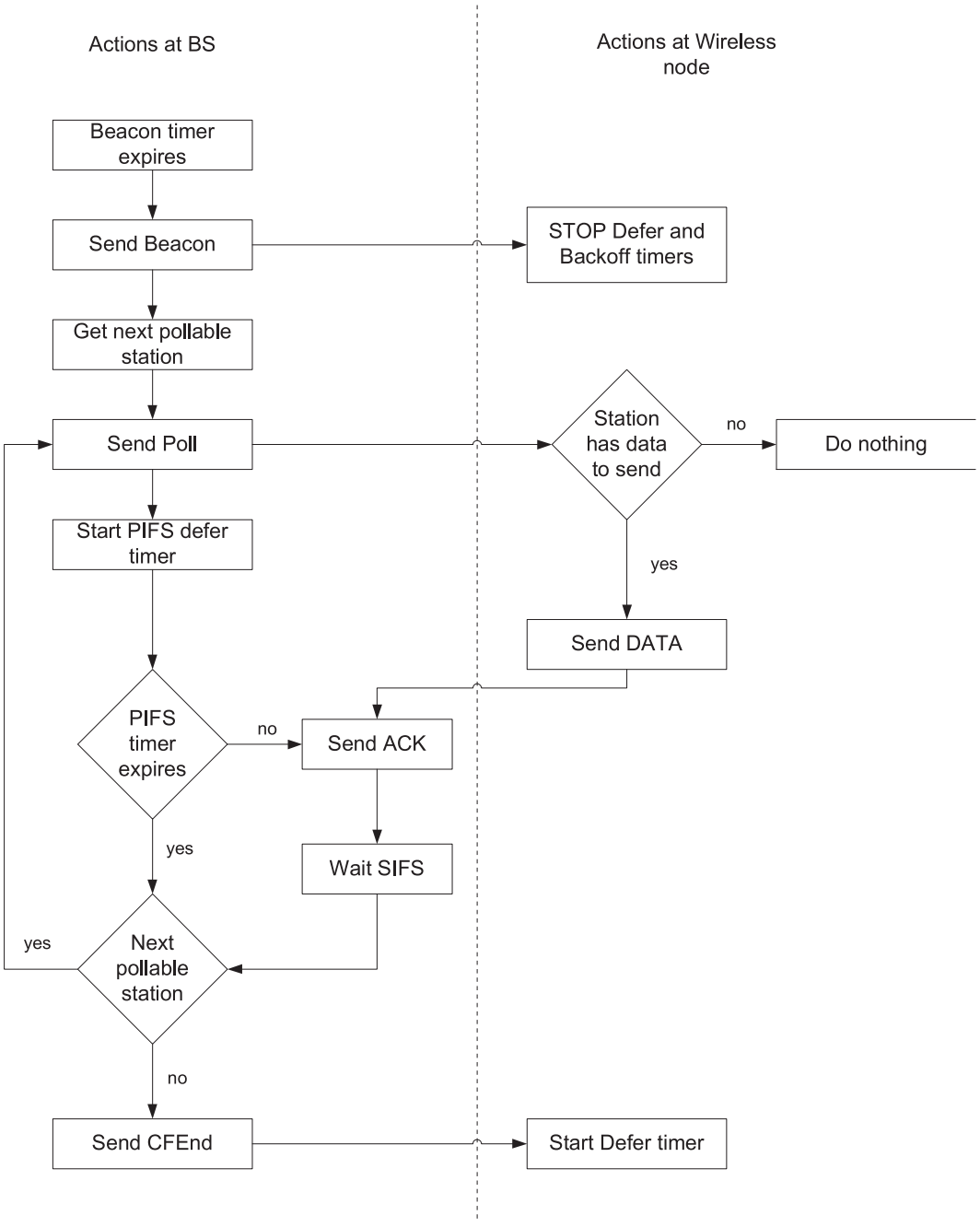


Figure 5.8: Program flow for original PCF toolbox

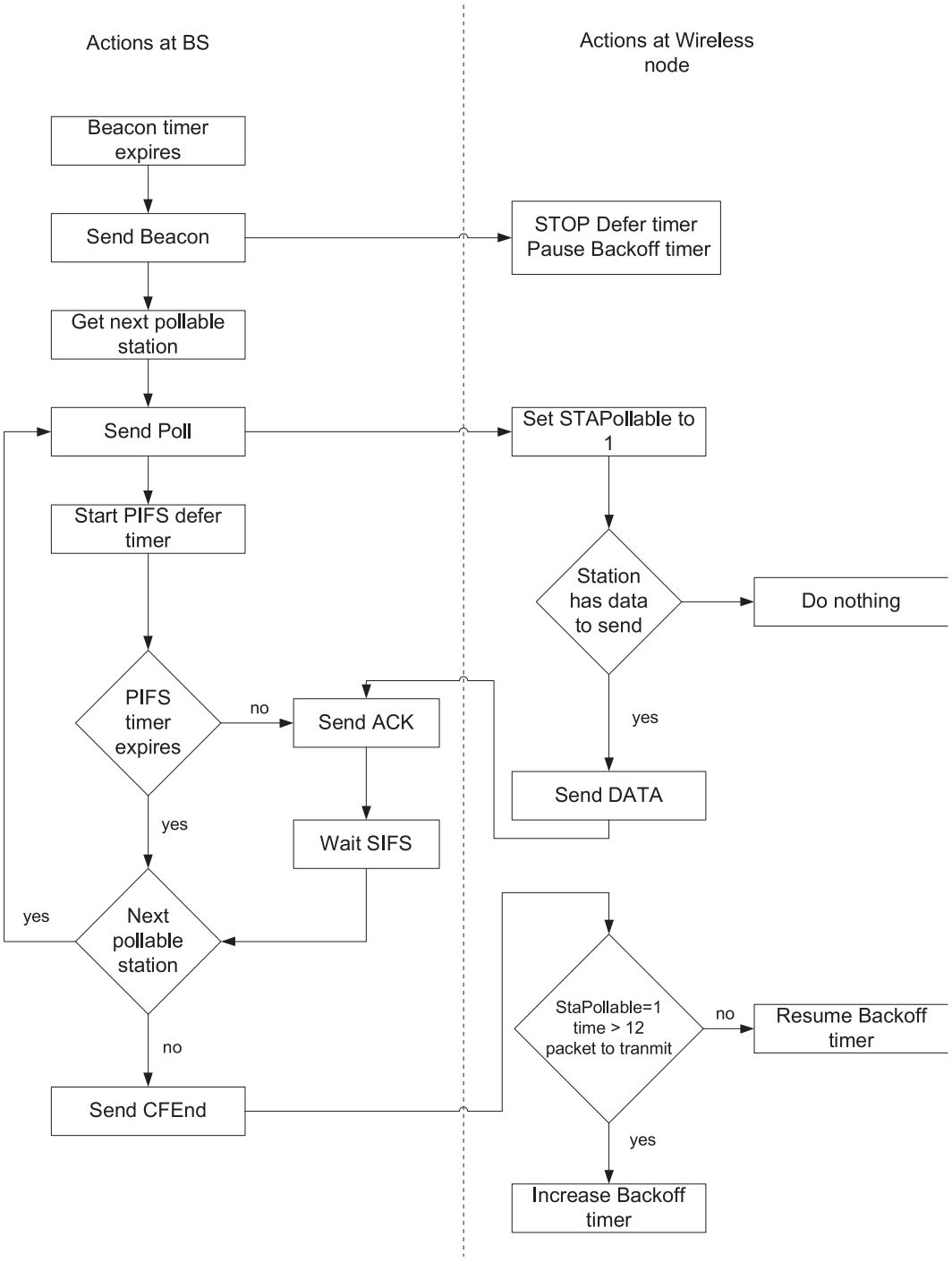


Figure 5.9: Program flow for modified PCF toolbox

tion frames to the BS. The solution to this problem was found by allowing pollable stations only to start transmitting during the CFP only after a predetermined period of time, to allow them to send their association frames during the CP. By trial and error it was found that all nodes associated with the AP within 12 seconds for 25 nodes; therefore, this value was used.

5.4.2 PCF simulation approach

The experimental setup used is the same as that outlined in section 5.2. Additionally, a Round Robin Polling (RRP) scheme is used to poll all stations during the CFP. Every station on the polling list is polled exactly once during the CFP. Enough time is left for the remainder of stations not on the polling list to transmit during the Contention Period (CP). Every beacon starts a CFP and the time interval between beacons will be the superframe interval.

5.4.3 PCF simulation parameters overview

The same simulation parameters were used as in section 5.2. The only additional parameters that need to be taken into consideration are the values of the superframe (SF) and the Contention Free Period Maximum Duration (CFPMaxdur). As the purpose of this project is not addressing the problem of adaptively finding the optimum values for the SF and CFPMaxdur, the values chosen will be enough for each station on the polling list to be polled once during a Contention Free Period (CFP). Accordingly, the value for the superframe will also be chosen as fairly large, so that there is enough time for nodes to operate in the CP and not too large that the throughput will deteriorate for CFP nodes.

5.4.4 ns2 Tcl script configuration

The *ns*2 PCF toolbox requires the AP to be set up as the polling master (this is done by using the AP's address and the command make-pc). Nodes which are only to operate in the CFP have to be manually added to the polling masters polling list (which is done by the command addSTA and specifying the node address).

5.4.5 PCF Throughput Simulation results

5.4.5.1 Goodput definition

The definition of goodput is the same as that used in section 5.3.1.1. The only difference between the DCF and PCF cases is that PCF has more control frames, such as Beacon, Poll and CFEnds which are not to be considered for goodput calculation. The unit of measurement remains bits per second (bps). It should be mentioned that the goodput can now be measured in three ways, for only pollable stations, for only DCF stations and as the combined goodput of all stations.

5.4.5.2 Factors on which the goodput will be dependent

With the parameters set to fixed values and with the consideration of only nodes operating in the CFP, the factors that will influence the goodput for PCF are the packet arrival rate, the packet size, the number of pollable nodes contending for the channel and the superframe size. As was mentioned in section 5.4.3 the value of CFPmaxdur was set to be large enough for all nodes to be serviced at least once in the CFP region. The units of measurement that will be used for the traffic arrival rate will be packets per second (pkts/s).

5.4.5.3 PCF goodput results and discussion

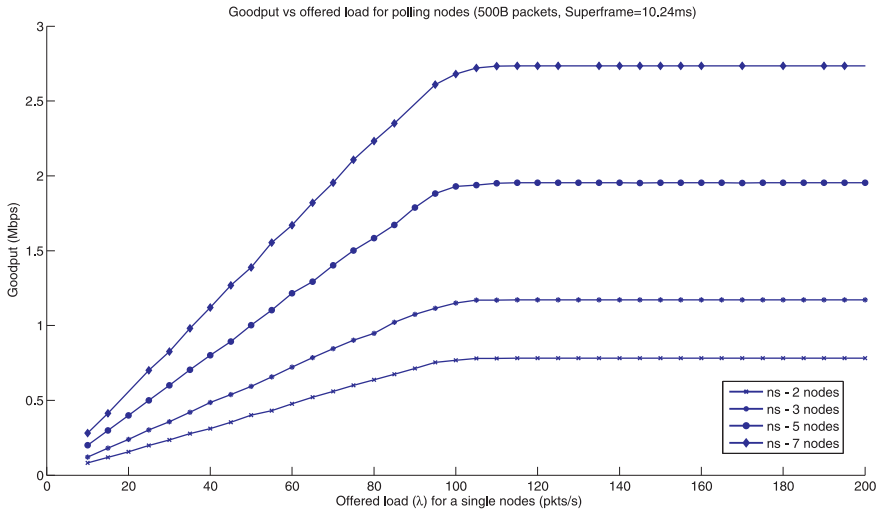


Figure 5.10: Average goodput versus the packet arrival rate for a variable number of pollable nodes, with no contention nodes (superframe=10.24 ms and constant 500B packet size)

As can be seen from the simulation results, if the parameters explained in Table 5.1 are fixed, the goodput is influenced by the following:

Number of nodes

The results clearly show the dependence of the goodput on the number of nodes. With the superframe chosen to be a fixed size and enough time for all nodes to be serviced in the superframe, it is obvious, as this is a deterministic protocol, that an increase in the number of nodes will increase the saturation throughput.

Packet size

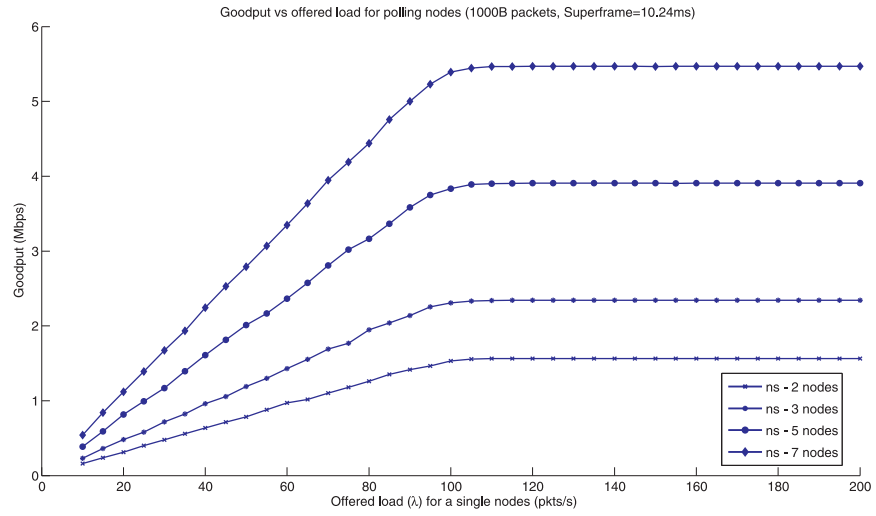


Figure 5.11: Average goodput versus the packet arrival rate for a variable number of pollable nodes, with no contention nodes (superframe=10.24 ms and constant 1000B packet size)

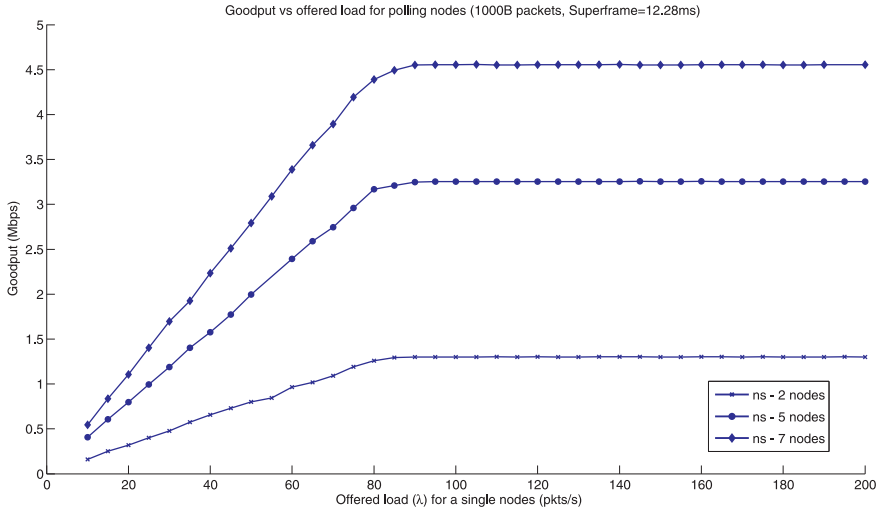


Figure 5.12: Average goodput versus the packet arrival rate for a variable number of pollable nodes, with no contention nodes (superframe=12.28 ms and constant 1000B packet size)

Larger packet sizes for PCF nodes will increase the saturation throughput value (compare figures 5.10 and 5.11 for the same superframe size). With enough time for all nodes to transmit in the CFP and a fixed superframe value, larger packet sizes will consume more of the available time for transmission in the superframe and increase the throughput.

Offered load

The throughput increases linearly until saturation is reached. The point of saturation is when the inter-packet arrival time becomes larger than the value used for the superframe period. If the superframe has a repetition interval of 10.24 ms, and the inter-packet arrival rate is increased above this, the throughput will saturate. The maximum service rate that a repetition interval of 10.24 ms can provide is 1/0.01024 = 97.66 packet per second, which implies that if the average packet arrival rate is larger than this, the throughput will saturate (see figures 5.10 and 5.11). In Figure 5.12 the superframe has a repetition interval of 12.28 ms, which implies that the maximum service rate is 1/0.01228=81.43 packets per second, accordingly the throughput will be less than for a superframe size of 10.24 ms with the same packet size (compare figures 5.11 and 5.12).

5.4.6 PCF packet delay results

5.4.6.1 Packet delay definition and additions to ns2 code

The packet delay definitions and additions to the *ns*2 code are the same as outlined those in section 5.3.2.1 and 5.3.2.2.

5.4.6.2 Simulation results and discussion

From Figure 5.13 the simulation results indicate that the only parameters that the average packet delay is dependant on are the packet arrival rate and the value of the superframe. The reason for this is that the service time for packets is deterministic, because the superframe delivers a fixed maximum service time that does not increase as the number of nodes increases; there is no change in the packet delay as the number of nodes increases. The same is true for the packet size. The packet delay starts to saturate as the maximum service rate is reached.

5.4.7 PCF buffer occupancy results

5.4.7.1 PCF buffer occupancy definition and additions to ns2 code

The buffer occupancy definitions and additions to the ns2 code are the same as those outlined in section 5.3.2.1 and 5.3.2.2.

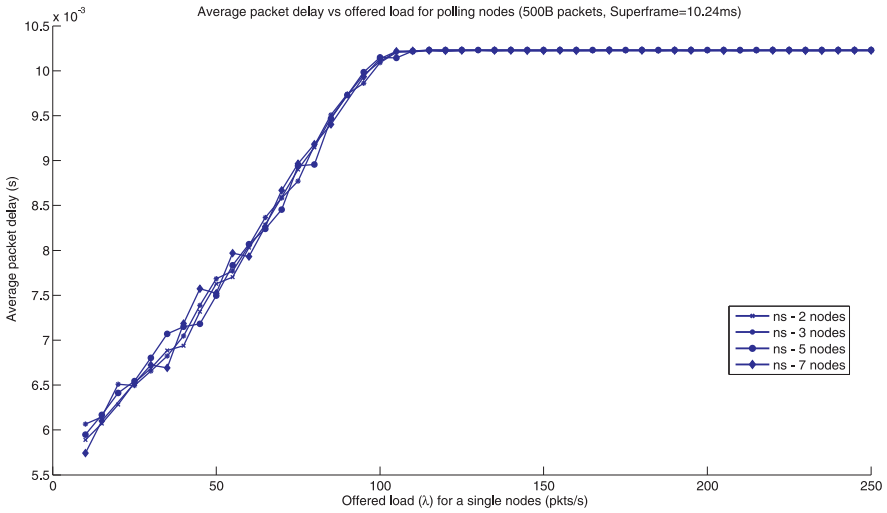


Figure 5.13: Average packet delay versus the packet arrival rate for a variable number of pollable nodes ($T_{SF} = 10.24ms$ and 500B packet sizes)

5.4.7.2 PCF buffer occupancy simulation results and discussion

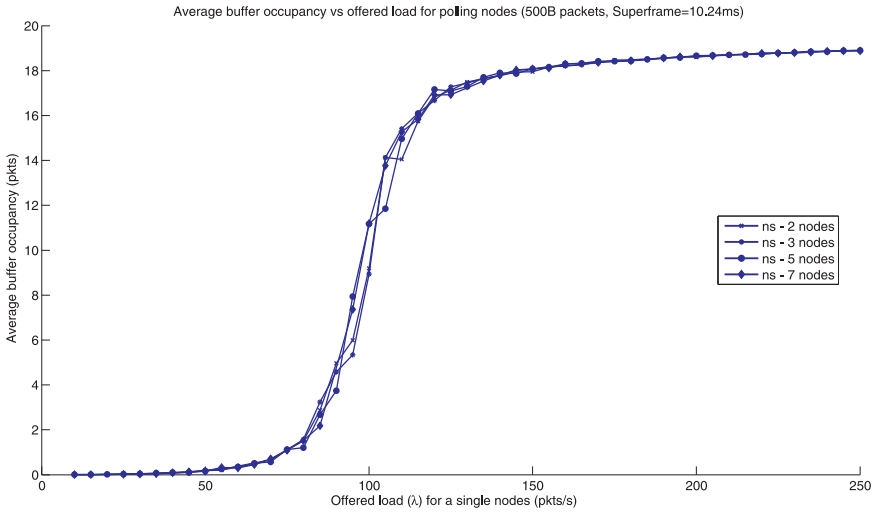


Figure 5.14: Buffer occupancy versus the packet arrival rate for a variable number of pollable nodes ($T_{SF} = 10.24ms$ and 500B packet sizes)

The simulation results for the average buffer occupancy are in agreement with those obtained in section 5.4.6.2; as the packet delay reaches saturation, the buffer occupancy starts increasing. The buffer occupancy is dependant only on the value of the

superframe repetition interval, as this determines the service rate and the packet arrival rate. This a blocking system, in that there is a finite buffer space of 20 packets, any new packet arrival will be discarded, as the results show.

5.5 Simulating the combined results of DCF and PCF in a superframe

5.5.1 Simulation approach

The simulation approach used and the definitions for throughput, delay and buffer occupancy is the same as that outlined in sections 5.3 and 5.4.

5.5.2 DCF nodes operating in conjunction with PCF nodes

5.5.2.1 DCF throughput in superframe structure, results and discussion

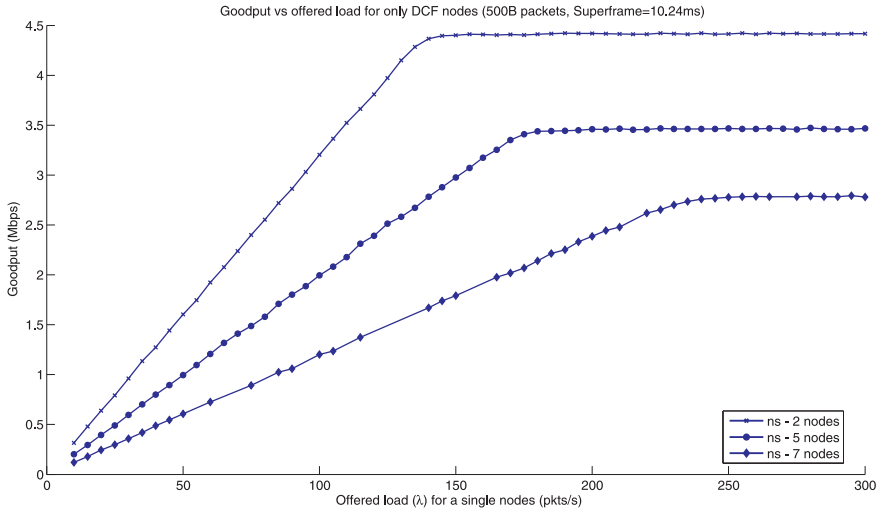


Figure 5.15: Average goodput versus the offered load for a total of 10 nodes, of which the number of pollable nodes is indicated on the figure (superframe=10.24ms and constant 500B constant size)

In Figures 5.15, 5.16 and 5.17 the number of nodes that operate in the CFP is indicated, which leads to the conclusion that the number of DCF nodes operating the CP is ten minus the number indicated on the figure.

It becomes evident from the figures that the more nodes are operating in the CFP, the less the throughput contributed by the CP nodes. This makes sense, because the more nodes that are operating in the CFP, the longer the CFP will be, and the less

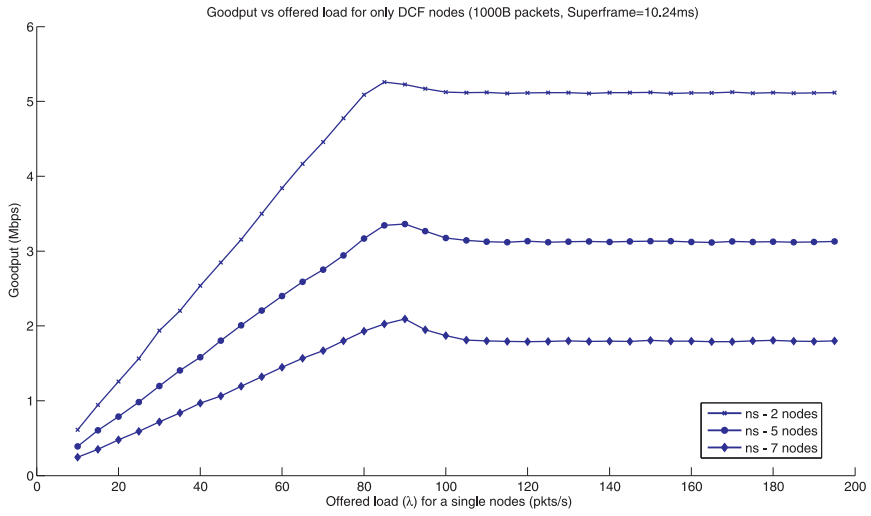


Figure 5.16: Average goodput versus the offered load for a total of 10 nodes, of which the number of pollable nodes is indicated on the figure (superframe=10.24ms and constant 1000B constant size)

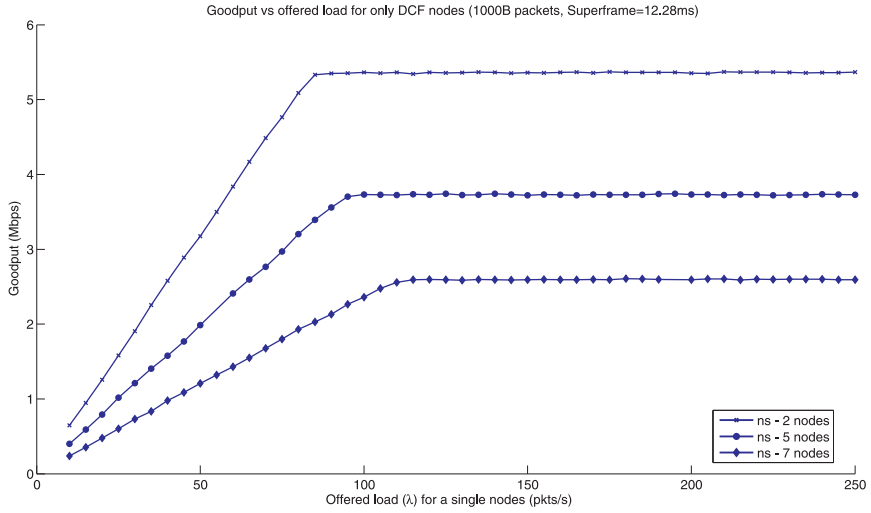


Figure 5.17: Average goodput versus the offered load for a total of 10 nodes, of which the number of pollable nodes is indicated on the figure (superframe=12.28ms and constant 1000B constant size)

the time available for contention nodes to transmit during the CP.

Notice by comparing Figure 5.16 and 5.17 for the case when the superframe is 10.24 ms, that the throughput of all nodes starts saturating roughly at the same packet arrival rate, whereas, comparing Figures 5.15 and 5.17, the shape of the curves is identical. The reason for this is that when the superframe value is 10.24 ms, the remaining time for the CP is too small for all nodes to transmit at least once, whereas, when the superframe is 12.28 ms, the remaining time in CP is enough for all DCF nodes to transmit at least once. This will have significant effect on the results of the buffer occupancy and packet delays, as will be shown in the sections to follow.

5.5.3 Combined throughput simulation results and discussion

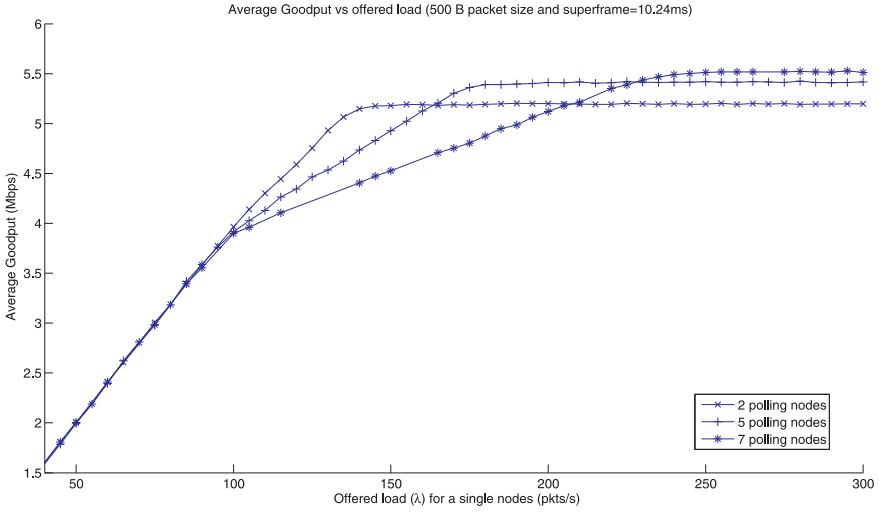


Figure 5.18: Comparison of the combined throughput versus the packet arrival rate for a total of 10 nodes, of which the number of CFP nodes and CP nodes are variable (superframe=10.24ms and constant packet size of 500B)

From Figure 5.18 it becomes evident that the combined average goodput is dependant on the value of the superframe repetition interval used and that is the reason for the curves not having the same linear increase as they initially had. At $\lambda = 97.66$ packets per second the service rate saturates, which is the reason for the decrease in the rate of increase of the throughput. This figure clearly shows the influence the superframe can have on the throughput.

From Figure 5.19, it seems that the throughput reaches saturation before the maximum service rate is reached for the nodes operating in the CFP, therefore the rate

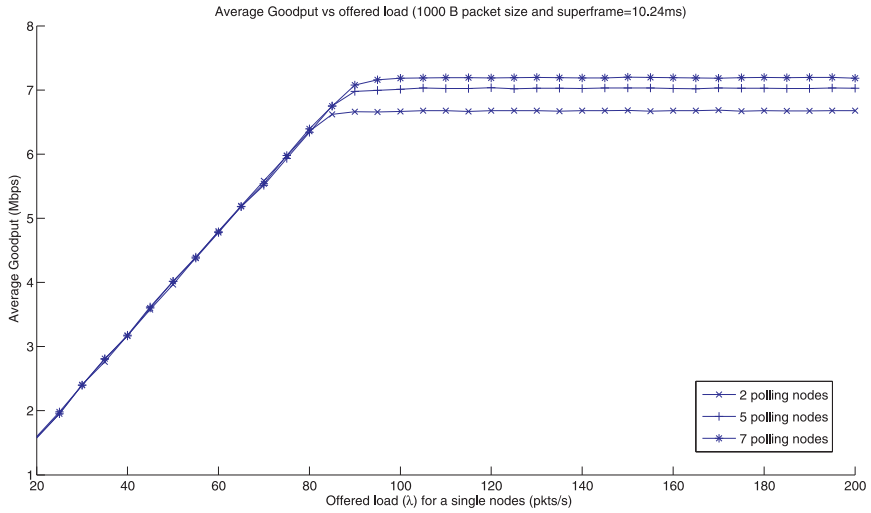


Figure 5.19: Comparison of the combined throughput versus the packet arrival rate for a total of 10 nodes, of which the number of CFP nodes and CP nodes are variable (superframe=10.24ms and constant packet size of 1000B)

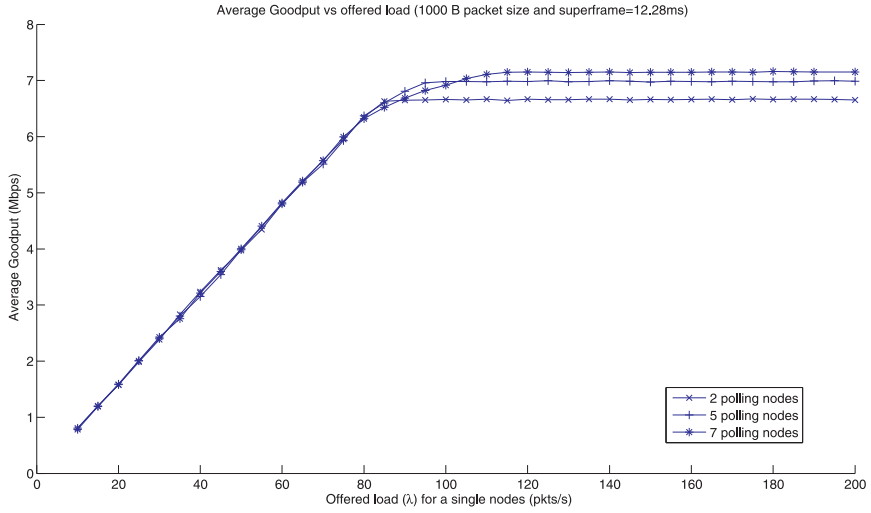


Figure 5.20: Comparison of the combined throughput versus the packet arrival rate for a total of 10 nodes, of which the number of CFP nodes and CP nodes are variable (superframe=12.28ms and constant packet size of 1000B)

of increase of the throughput remains linear until saturation is reached.

It should be mentioned here that it can be observed that the combined throughput of DCF and PCF nodes shows the throughput to be the result of superposition of the throughputs obtained by DCF and PCF nodes individually. That is, the saturation throughput when 5 nodes are operating in the CP for a 1000B packet size and a superframe of 12.28 ms (see Figure 5.17) is 3.733 Mbps. For the 5 PCF nodes only operating in the CFP for the same network parameters and configuration, the throughput obtained is 3.253 Mbps (see Figure 5.12). The combined throughput obtained is 7.709 Mbps, which is consistent with the combined throughput obtained in Figure 5.20 with a value of 6.984 Mbps. It does make sense, because the measurement for throughput for both the CFP and CP is in bits per second, and reducing the time available for transmission will scale the throughput accordingly. Using this method the results are consistent for Figure 5.18 and 5.19. This finding will be discussed in further detail in Chapter 8.

5.5.4 DCF packet delay in a superframe structure (Results and discussion)

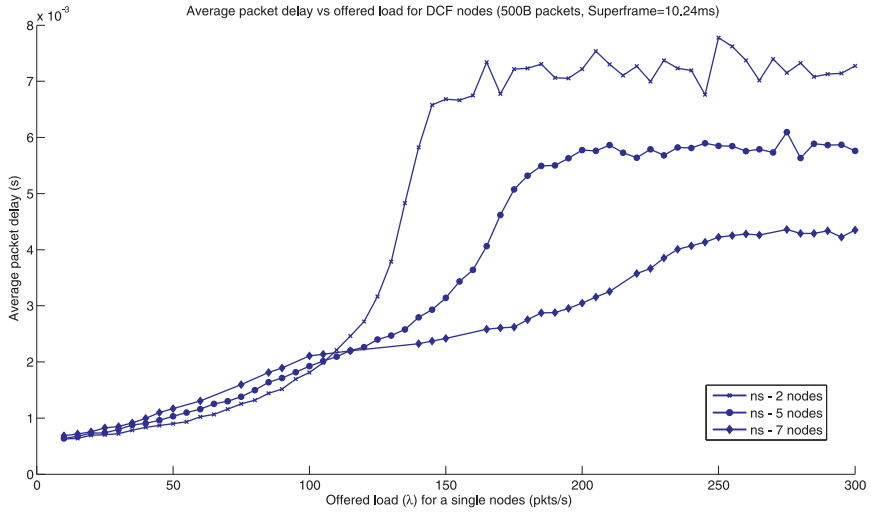


Figure 5.21: Comparison of the average DCF packet delay versus the packet arrival rate for a total of 10 nodes, of which the number of CFP nodes and CP nodes are variable (superframe=10.24ms and constant packet size of 500B)

The results obtained in figures 5.21,5.22 and 5.23 are in agreement with those in section 5.5.2.1, because the figures indicate that the packet arrival rate at which saturation occurs for the throughput and the packet delay agree.

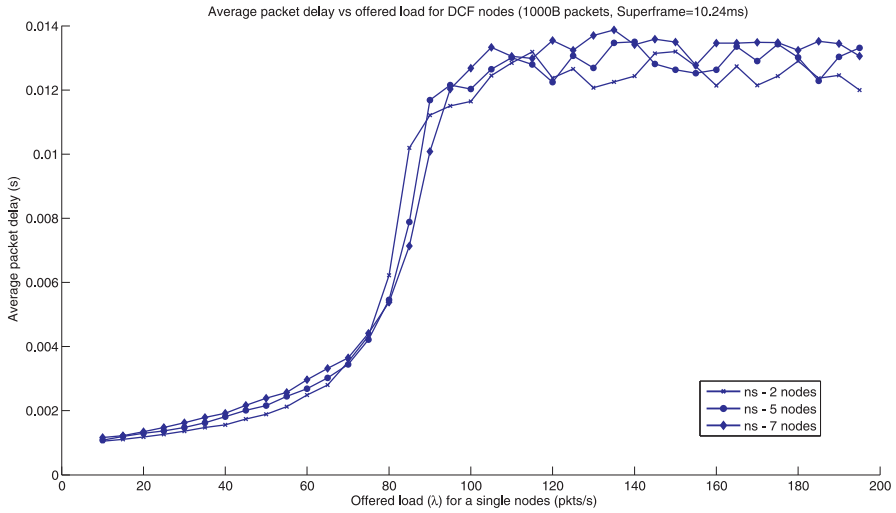


Figure 5.22: Comparison of the average DCF packet delay versus the packet arrival rate for a total of 10 nodes, of which the number of CFP nodes and CP nodes are variable (superframe=10.24ms and constant packet size of 1000B)

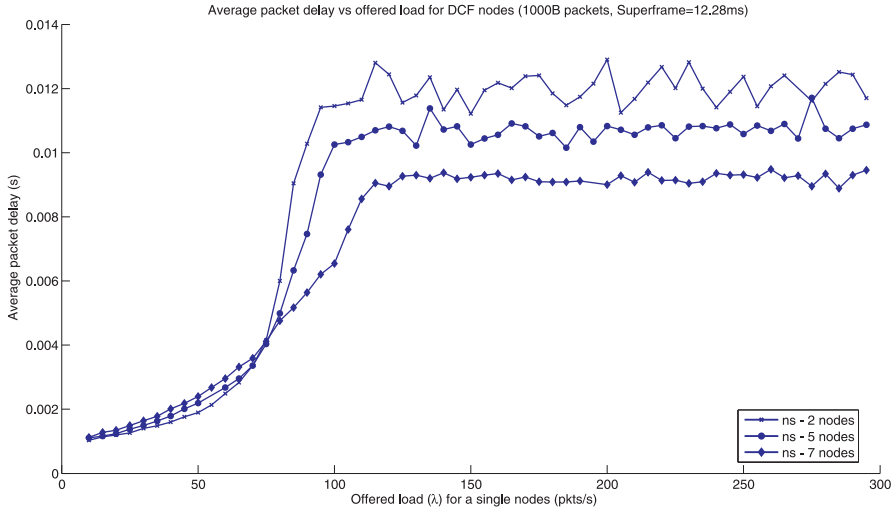


Figure 5.23: Comparison of the average DCF packet delay versus the packet arrival rate for a total of 10 nodes, of which the number of CFP nodes and CP nodes are variable (superframe=12.28ms and constant packet size of 1000B)

Figure 5.22 proves the finding that before maximum service rate is reached for the PCF nodes, the packet delay saturates. This explains the findings of a superframe that is too small for all nodes to be able to transmit exactly once.

The finding here is also that the packet delay does significantly increase as the CFP consumes transmission from the CP nodes. For example, from Figure 5.21 the delay for 5 CFP and 5 CP nodes is 5.895 ms, and from Figure 5.4 the delay for 5 DCF nodes is 3.76 ms. This does, however make sense if a CP node has a packet to transmit and is counting down its backoff timer, and then during this process a CFP starts, the backoff timer is paused and resumes counting down from the same value. The packet delay will be increased by the duration of the superframe. The results for Figure 5.22 and 5.22 are also consistent with these findings. This occurrence will be further discussed in Chapter 8.

5.5.5 DCF buffer occupancy in a superframe structure results and discussion

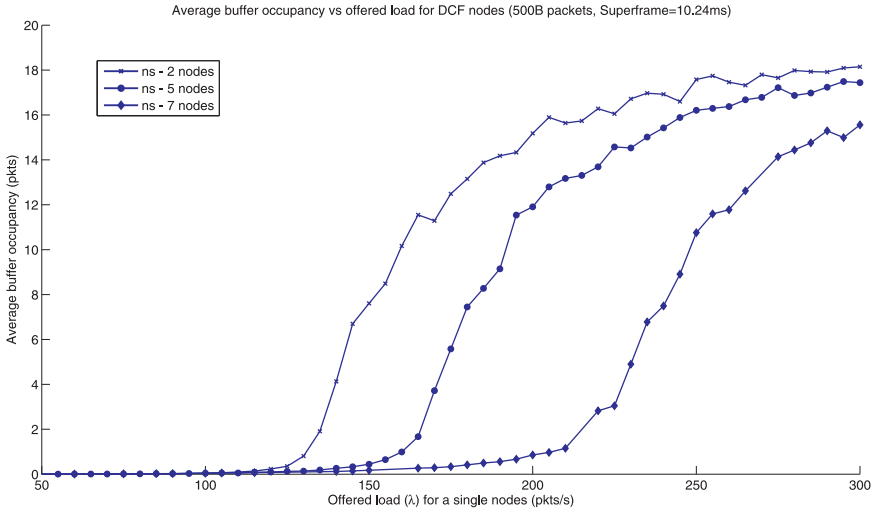


Figure 5.24: Comparison of the average DCF buffer occupancy versus the packet arrival rate for a total of 10 nodes, of which the number of CFP nodes and CP nodes are variable (superframe=10.24ms and constant packet size of 500B)

The results in this section for the DCF buffer occupancy are in agreement with the results obtained for the DCF packet delay. Notice that only when the packet delay

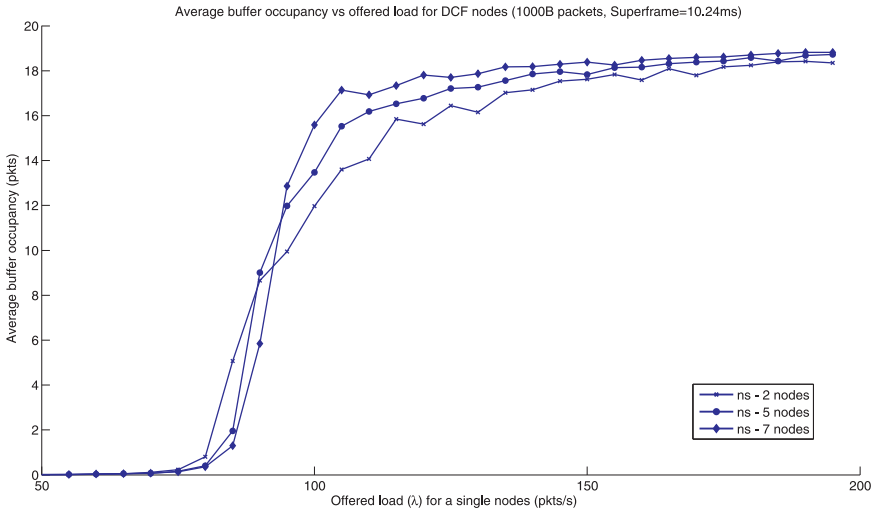


Figure 5.25: Comparison of the average DCF buffer occupancy versus the packet arrival rate for a total of 10 nodes, of which the number of CFP nodes and CP nodes are variable (superframe=10.24ms and constant packet size of 1000B)

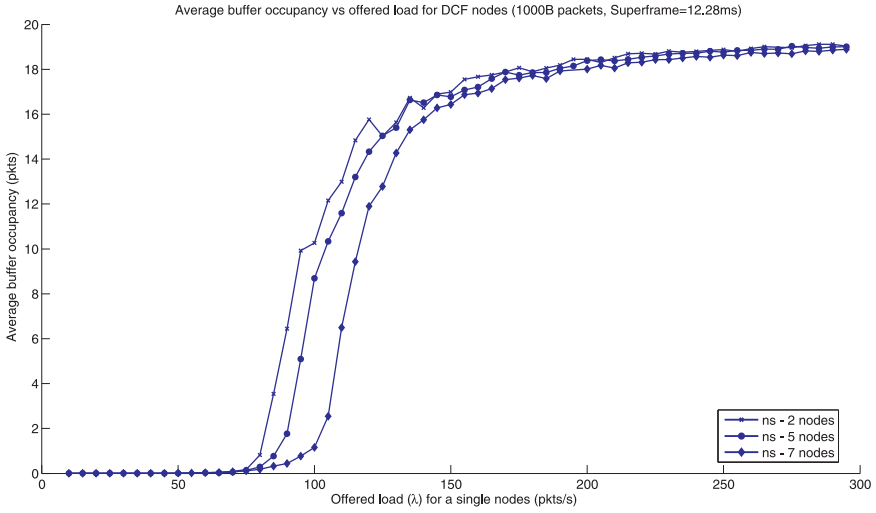


Figure 5.26: Comparison of the average DCF buffer occupancy versus the packet arrival rate for a total of 10 nodes, of which the number of CFP nodes and CP nodes are variable (superframe=12.28ms and constant packet size of 1000B)

reaches its saturation value does the buffer occupancy start to increase.

The results also show that fewer nodes operating in the CP obviously decreases the rate at which the average buffer occupancy increases (from Figures 5.24 and 5.26).

The results are also consistent in that the maximum average value that is reached never exceeds the maximum buffer occupancy parameter as set in *ns*2.

Again, notice the effect the value of the superframe can have if too small a value is chosen, by comparing Figures 5.25 and 5.26.

5.5.6 Summary

In this chapter the statistical and configuration approaches to modelling of the protocol were discussed. Some simulation results were also shown and compared with the theory of CSMA and polling protocols. The results proved to be in agreement, but further mathematical investigation will have to be done to prove this, as will be shown in the chapters following. The small additions to the *ns*2 code is also included in this section.

Chapter 6

DCF mathematical model

This chapter presents a discussion of a mathematical model that was created to predict the DCF behaviour of an IEEE 802.11 network. The model will provide performance predictions for throughput, packet delay and buffer occupancy.

6.1 Motivation for mathematical model

It is imperative to create a mathematical model to provide theoretical verification of simulation results. The results obtained from the model will also aid in discerning which communication protocols are better suited to which circumstances.

6.1.1 DCF mathematical model literature overview

In order to accurately model a network, it is imperative to consider both saturated and unsaturated traffic conditions. Previous work on the performance of the 802.11 MAC has focused primarily on throughput under saturated conditions [Bia00]. This work was also conducted by consideration of unsaturated conditions such as [DM07], which only considers buffers with one packet. Other analytical models have been developed [ST07] that rely on statistical methods, but fail to model certain events such as pre and post backoff mechanisms. The most comprehensive model that has surfaced from study of literature is Garetto's model [CG05]; which considers criteria such as the buffer occupancy and the number of active transmitting stations, and which models the maximum number of states. Consequently, this model was chosen to be exploited as the basis of further analysis.

6.2 Queueing Theory

6.2.1 Background

Queueing theory is the study of waiting lines, better known as queues. Agner Krarup Erlang published the first paper on queuing theory in 1909. Since then it

has become a major academic field on its own. Another pivotal moment was in 1953, when David G. Kendall introduced his A/B/m/k queueing notation, where A describes the arrival process, B describes the service process, m the number of servers for the queue and k is used as a subclass descriptor, for example the buffer size. The possible options for the arrival (A) and service (B) processes are Markovian (M), Deterministic (D), General (G), General Independent (GI), Erlang (E) and Hyper-exponential (H).

6.2.2 Basic queueing principles

A basic queueing model is shown in Figure 6.1. Every queueing system consists of an input from a source that generates data at a specific rate λ, the queue (which will be discussed below), and the output to a server or some other process with a service rate of μ.

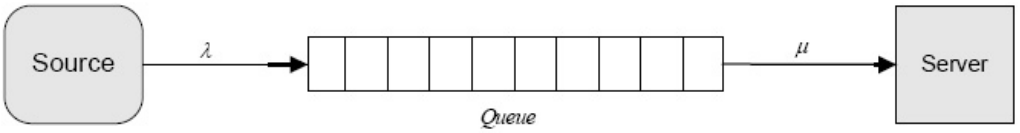


Figure 6.1: Basic queueing network

The parameters that characterize the queueing system are discussed below:

Source population

The source population describes the source that connects to the queue. For a WLAN, the number of stations in a network is limited, while the number of events can be limited or unlimited. The size of the arriving population may be characterised as infinite if the number of potential packets from external sources is large compared to the packets already present in the system, or it may be finite in the event that the arrival rate is much lower than the system capacity. For a system that is constrained by a finite service rate, the size of the arriving population has an impact on the queueing results. Therefore, in the practical application of WLANs, if each source node always has a new packet to transmit (an infinite source population), the queueing system will be saturated with traffic. The mathematical model for a saturated queueing system is significantly simplified, as indicated in section 6.3. However, for the unsaturated case, the mathematical model explained in section 6.5 is more involved because the arrival process has to consider the number of packets already in the system, as well as the number of idle stations.

Arrival patterns

Customers may arrive at a queueing system in either a regular pattern or in a random fashion. When customers arrive at a queue at a fixed rate then their arrival is described only by the rate of arrival. However, if customers arrive in a random fashion, a fitting statistical model is required for the arriving pattern in order to perform an accurate mathematical queuing analysis. A Poisson distribution is the most commonly used random arrival pattern, and is therefore also the model employed in this project.

Behaviour of arriving customers

If the service capacity of a queueing system is insufficient for the given source population (typically characterised by finite buffer space and all servers being busy and operating at full capacity), the system becomes full or saturated. A new packet that arrives at a full system is discarded without entering the system, and the queue is referred to as a blocking system. On the other hand, there are systems that are modelled using infinite queues, that is, queues with unlimited capacity. As for this project, a wireless node has finite buffer space, making it a blocking system.

Physical layout and number of servers in the system

The service facility may comprise of one or more servers; the server units can also be connected for either serial or parallel processing. For the purpose of this project, there is a single server, namely the Wireless Access Point (interchangeably referred to as the Base Station).

Queueing discipline

Queueing discipline refers to the order in which packets are fetched and subsequently serviced from the queue. Typical disciplines are: First in First out (FIFO), Last in First out (LIFO), priority, processor sharing and random. FIFO does not assign priorities and serves packets in the order which they arrive. Because of its simplicity this is the discipline used.

Service distribution

Similar to arrival patterns, if all the customers require the same constant service time, the service process becomes deterministic.

6.2.3 Modeling of queueing systems

Modelling of queueing systems is usually based on Markov's chain theory. A Markov process is memoryless - this implies that if we observe an event at a certain point in time, the time of the next arrival is not affected by the interval of time

since the previous arrival. In other words, the process starts afresh at the time of observation and has no memory. This memoryless property proves to be useful for statistical analysis of networks, especially since it permits the omission of prior activities in network analysis.

Another key concept is that of resource utilization, which is a measure of how busy the server is. Utilization is the fraction of the time a server is busy providing a service conditioned by the time available to provide that service. Utilization (ρ) is defined as

$$\rho = \frac{\lambda}{m\mu} \tag{6.2.1}$$

where λ is the packet arrival rate (in *packets/second*), m the number of servers and μ service rate (in *packets/second*). ρ is dimensionless and should be less than unity for the system to be stable, and to cope with the service demand.

The Poisson distribution is pivotal to modelling queueing systems. In most elementary queueing systems the inter-arrival and service times are assumed to be exponentially distributed or, as will be explained, can be modelled as an exponential distribution. Essentially, a Poisson process is a counting process of the number of events occurring during a time interval (0, t), by using the well-known Poisson distribution given below.

$$P_n(t) = \frac{(\lambda t)^n}{n!} e^{-\lambda t} \tag{6.2.2}$$

In the equation above, λt is the mean of the Poisson random variable and it represents the probability distribution of the number of occurrences during a time interval t.

The exponential distribution and Poisson process closely mirror each other. If the inter-arrival times in a point process are exponentially distributed, the number of arrivals in a specific time period would be given by the Poisson process.

6.2.4 Little's theorem

An important principle is that of Little's theorem, discovered in 1968 by J.D.C Little. It states that if a queueing system is in steady state, the average number of entities (N) is equal to the product of the average arrival rate (λ) and the average time (T) it takes for each entity to be serviced, as given by the equation below.

$$N = \lambda T \tag{6.2.3}$$

Given that the system is stable and there is adequate buffer space, this theorem holds true for all types of queueing systems. The proof of this theorem can be found in [Hok-97].

This succinct introduction to queueing theory should provide a sufficient foundation to explore the mathematical model. Additional theory will be explained as the chapter progresses.

6.3 Modeling a DCF saturated network

One of the first queueing models to model an IEEE 802.11 network using the Distributed Coordination Function was presented by Giuseppe Bianchi [Bia-00]. He modelled the behavior of a single station using a Markov model. It was created to determine the maximum saturation throughput (goodput) for a saturated network, which implies that as soon as a node successfully transmits a packet, a new packet is immediately available. The analysis will be divided into two parts. First, the behavior of a single station is studied by using a Markov model to obtain the stationary packet transmission probability τ for any randomly chosen time slot. Secondly, the throughput is determined for a wireless network by using the computed τ value. This model serves as a good background for the following sections in this chapter and the model that was used.

6.3.1 Model assumptions

The main assumptions are as follows:

- A fixed number of competing stations are accessing the same wireless channel. Stations always have data to send and, therefore, they operate under saturated conditions.
- There are no hidden terminals.
- Stations are equally likely to access the channel.
- The communication channel is error-free.

6.3.2 Packet transmission probability

A discrete and integer time scale is used, where t and $t+1$ correspond to the beginning of two consecutive time slots. At the beginning of each time slot the backoff counter decrements. As illustrated in 6.2, if the backoff timer decrements normally and there are no packet transmission, the time between two consecutive slots is the constant slot time size (σ). For example from Figure 6.2, when station B decrements it's backoff timer from 2 to 1, which is indicated by σ. If there is a packet

transmission during two consecutive slot times, then that specific slot time will be longer. For example from Figure 6.2, when station B decrements it's backoff timer from 5 to 4, station A transmits a packet. This is referred to as the variable time interval. $b(t)$ represents a stochastic process for the backoff timer of a single specific node. Please refer to Chapter 3 for an overview of the backoff mechanism of the IEEE 802.11 standard.

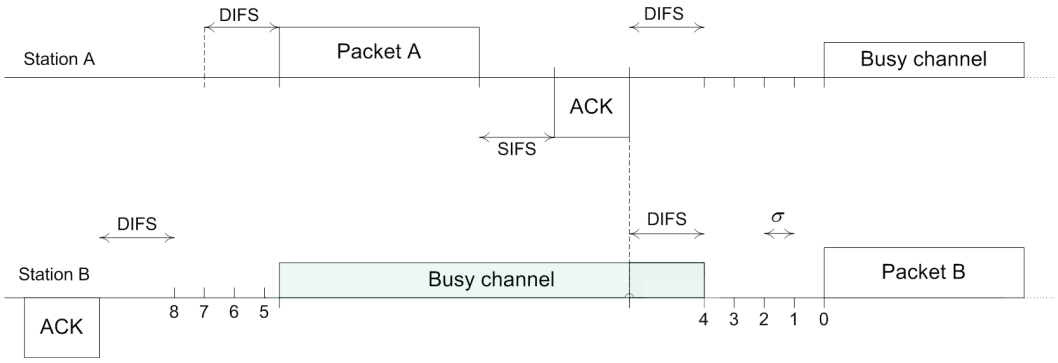


Figure 6.2: Example of a basic access mechanism for DCF with 2 stations

The value of the backoff timer of each station depends on previous transmissions, and on the previous number of retransmissions the packet at the front of the queue has suffered, making this process non-Markovian.

The minimum contention window (CW_{min}) is defined as $W = CW_{min}$. Let m be the "maximum backoff stage", such that $CW_{max} = 2^m \cdot W$. The notation is adapted to $W_i = 2^i \cdot W$, where i $\in$ (0, m) is called the "backoff stage" (referred to in section 3.7.1 as the backoff counter). The backoff stage starts with an initial value of zero and with every failed transmission attempt it is increased by one. $s(t)$ represents a stochastic process for the backoff stage in the range (0,m) of a station for time t. The process of choosing a new backoff value is modelled by using a uniformly distributed random process in the range (0, W_i). After a successful packet transmission the node returns to backoff stage zero.

A cardinal approximation for this model is that any packet has a fixed collision probability, independent of the number of retransmissions. That is, any packet transmission has the same probability of resulting in a collision. This probability is called the conditional collision probability p.

With the assumptions of independence and a constant value for p, it is possible to model the process $(s(t), b(t))$ with discrete time instants as a Markov process.

Table 6.1: Summary of transition probabilities for Bianchi's model

s_i	s_f	$P(s_i, s_f)$	Condition
i,k	$i,k-1$	1	$1 \leq k \leq (W_i - 1)$ $0 \leq i \leq m$
$i,0$	$0,k$	$\frac{(1-p)}{W_0}$	$0 \leq k \leq (W_i - 1)$ $0 \leq i \leq m$
$i-1,0$	i,k	$\frac{p}{W_i}$	$0 \leq k \leq (W_i - 1)$ $1 \leq i \leq m$
$m,0$	m,k	$\frac{p}{W_0}$	$0 \leq k \leq (W_i - 1)$

Table 6.3.2 summarises the transition probabilities from the Bianchi Markov chain model in Figure 6.3. In Table 6.3.2 s_i is the initial state and s_f the final state, $P(s_i, s_f)$ is the transition probability and the 4th column indicates the states for which this is possible. In the columns for s_i and s_f in Table 6.3.2, and in Figure 6.3, the variables i and k respectively describe the backoff stage and the backoff timer value. The first equation accounts for the transition probability when the node is in backoff. At the beginning of each time step the backoff timer decrements; this transition probability is one, because the system has only one state to go to. The second equation accounts for the time instant after a successful packet transmission, when the node returns to backoff stage 0, a new backoff counter value is uniformly chosen in the range $(0, W_0 - 1)$ and the probability for a successful packet transmission $(1 - p)$. The last two entries in the table compensate for unsuccessful packet transmissions. The third equation states that if the backoff timer's value is zero, the probability that the transmission will result in a collision is p, and W_i is for the new backoff value to be chosen. The last equation compensates for the system reaching its maximum backoff stage, when the backoff counter will not be increased any further.

Note that this system has an infinite retry limit. That is, if the maximum backoff stage is reached, the packet will not be discarded or dropped. The system will persist in trying to transmit the packet, until it is successful.

Let the stationary distribution of the chain with a simplified notation be defined as $b_{i,k} = lim_{t \to \infty} P\{s(t) = i, b(t) = k\}$ for $i \in (0, m), k \in (0, Wi - 1)$. First note method of relation

$$b_{i-1,0} \cdot p = b_{i,0} \qquad \rightarrow \quad b_{i,0} = p^i \cdot b_{0,0} \quad \text{for} \quad 0 < i < m \tag{6.3.1}$$

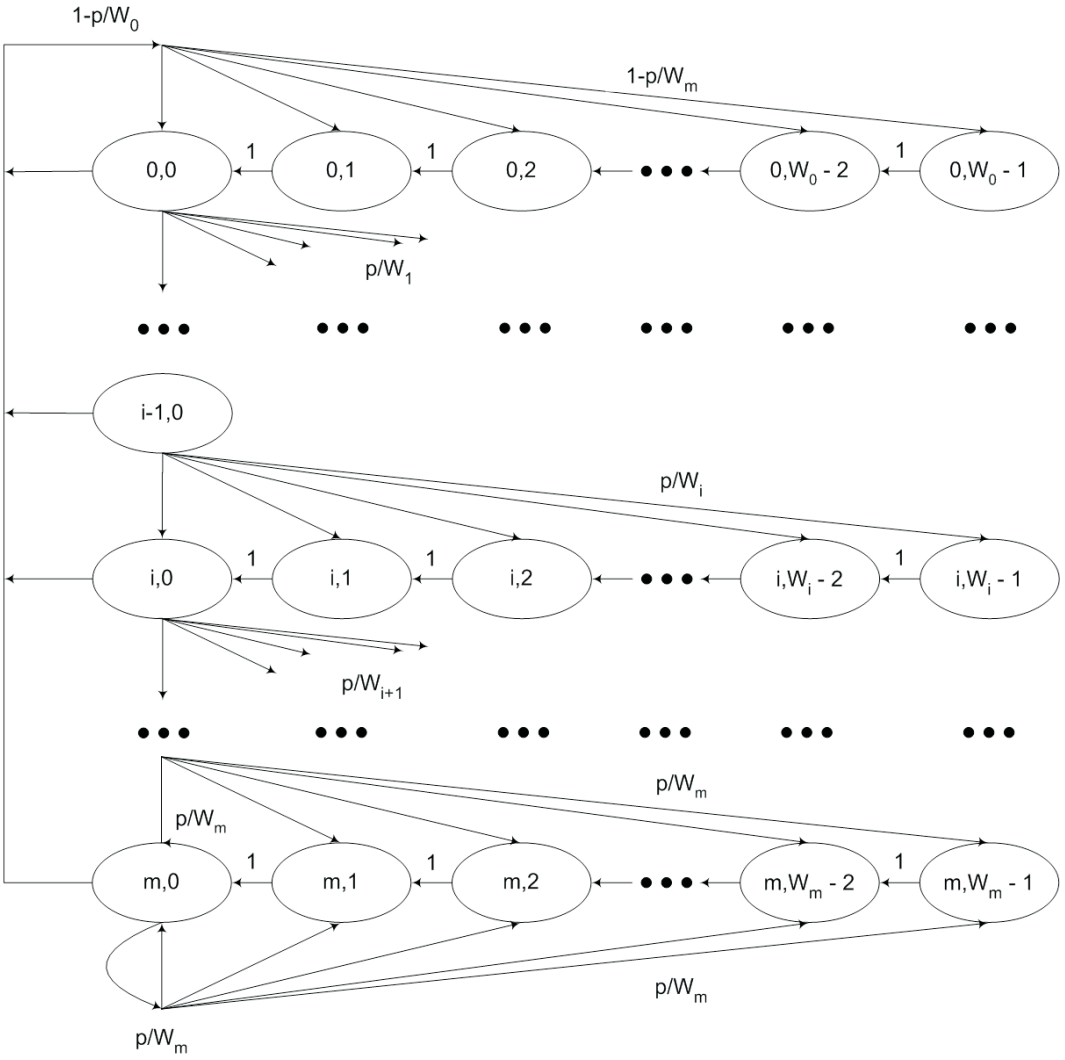


Figure 6.3: Bianchi Markov chain model

The equation states that the probability of reaching the next backoff stage is dependant solely on the conditional collision probability (p), which is modelled as a geometric series.

As the final backoff stage can be reached only from stages $m-1$ and m,

$$b_{m,0} = p \cdot b_{m-1,0} + p \cdot b_{m,0} \qquad \rightarrow \quad b_{m,0} = \frac{p^m}{1-p} \cdot b_{0,0} \tag{6.3.2}$$

where the relation in equation 6.3.1 was used to simplify equation 6.3.2.

To compensate for the backoff chain process, finding the probability of a node being at a specific backoff timer value is determined by the following relation

$$b_{i,k} = \frac{W_i - k}{W_i} \cdot b_{i,0} \quad i \in (0,m), k \in (0, W_i - 1) \tag{6.3.3}$$

This equation compensates for the fact that the lower the value of backoff timer, the higher the probability of the system being in that state. Higher randomly chosen backoff values still have to pass through lower ones to reach zero.

The state probabilities can now be written as

$$b_{i,k} = \frac{W_i - k}{W_i} \cdot \begin{cases} (1-p) \cdot \sum_{j=0}^{n} b_{j,0} & i = 0 \\ p \cdot b_{i-1,0} & 0 < i < m \\ p \cdot (b_{m-1,0} + b_{m,0}) & i = m \end{cases} \tag{6.3.4}$$

The first line in equation 6.3.4 accounts for the probability of finding the system in state $b_{0,0}$. This is determined as the combination of the probability of a successful transmission $1-p$, and the sum of all the probabilities for $b_{i,0}$, as these are the only states from where transmission is possible, and where $0 \leq i \leq m$. That is $(1-p) \cdot \sum_{i=0}^{m} b_{i,0} = b_{0,0}$.

Thus, by method of relations, equation 6.3.2, 6.3.3 and 6.3.4 are combined to relate all states to $b_{0,0}$ and p. The normalization property (refer to section 6.5.4 for an explanation) is used to sum all states to one, which gives the following equation as

$$1 = \sum_{i=0}^{m} \sum_{k=0}^{W_i-1} b_{i,k} = \sum_{i=0}^{m} b_{i,0} \sum_{k=0}^{W_i-1} \frac{W_i - k}{W_i} = \sum_{i=0}^{m} b_{i,0} \frac{W_i + 1}{2} = \frac{b_{0,0}}{2} \left[W \left(\sum_{i=0}^{m-1} (2p)^i + \frac{(2p)^m}{1-p} \right) + \frac{1}{1-p} \right] \tag{6.3.5}$$

Simplifying the above expression gives

$$b_{0,0} = \frac{2(1-2p)(1-p)}{(1-2p)(W+1) + pW(1-(2p)^m)} \tag{6.3.6}$$

The probability that a station transmits during a randomly chosen time slot is defined as τ. Stations can only transmit if their backoff counter value is zero, therefore

$$\tau = \sum_{i=0}^{m} b_{i,0} = \frac{b_{0,0}}{1-p} = \frac{2(1-2p)}{(1-2p)(W+1)+pW(1-(2p)^m)} \tag{6.3.7}$$

Unfortunately τ depends on the conditional collision probability p, which is still unknown. To find p, it is sufficient to note that the probability of a packet collision, if a single random station out of a total of n nodes is attempting to transmit in a specific time slot, will be the probability that any of the remaining $n-1$ stations attempts to transmit in that particular time slot. An assumption is made that the states of all stations are fundamentally independent, implying that each transmission "sees" the system in the same state, which would be in steady state. Using this assumption, the transmission probability for any packet is τ, therefore

$$p = 1-(1-\tau)^{n-1} \tag{6.3.8}$$

Here $(1-p)$ is the probability of a successful transmission and the probability of no packet transmission is $(1-\tau)$. The probability that a packet will result in a successful transmission is therefore $(1-p) = (1-\tau)^{(n-1)}$ for n stations, which gives equation 6.3.8.

Equations 6.3.7 and 6.3.8 have to be solved by numerical methods because they represent a non-linear system with two unknowns τ and p.

6.3.3 System throughput

To analyse the system throughput it is necessary to monitor activities during randomly chosen time slots. Let P_{tr} be the probability that at least one packet is transmitted during a randomly chosen time slot. Since n stations are contending for the channel, and each station transmits with probability τ,

$$P_{tr} = 1-(1-\tau)^n \tag{6.3.9}$$

The probability that a packet transmission is successful is defined as P_S. It will be dependant on the probability that a single station transmits a packet and no other stations will transmit during this time slot, conditioned by the probability that a transmission does take place.

$$P_s = \frac{n\tau(1-\tau)^{n-1}}{P_{tr}} \tag{6.3.10}$$

Let T_P be the aggregate packet throughput, that is, the fraction of time the channel is occupied by payload packets. T_P can be expressed as

$$T_P = \frac{E[payload \quad information \quad transmitted \quad per \quad time \quad slot]}{E[length \quad of \quad time \quad slot]} \tag{6.3.11}$$

The probabilities that the channel is occupied by a successful transmission (Π_s), a collision (Π_c), or an idle slot (Π_σ), are computed by respectively:

$$\Pi_s = n\tau(1-\tau)^{n-1} = P_{tr} \cdot P_s \tag{6.3.12}$$

$$\Pi_\sigma = (1-\tau)^n \tag{6.3.13}$$

$$\Pi_c = 1 - \Pi_\sigma - \Pi_s \tag{6.3.14}$$

Finally, the aggregate packet throughput T_p is determined (expressed in packets/s). This is given by,

$$T_p = \frac{\Pi_s}{\Pi_s T_s + \Pi_s T_s + \Pi_\sigma \sigma} \tag{6.3.15}$$

Here T_s is the average time the channel is sensed as busy with a successful packet transmission, T_c is the average time the channel is occupied by a packet collision and σ is the duration of an idle time slot.

As we only considered the Basic Access mechanism, and not the RTS/CTS access mechanism, we define

$$T_s = H + E[P] + SIFS + \delta + ACK + DIFS + \delta \tag{6.3.16}$$

$$T_c = H + E[P^*] + DIFS + \delta \tag{6.3.17}$$

with $H = PHY_{hdr} + MAC_{hdr}$, where PHY_{hdr} and MAC_{hdr} being the physical and

MAC packet headers, and δ representing the propagation delay, $E[P]$ represents the time necessary to transmit the average packet payload size and $E[P^*]$ the size of the longest packet collision. As in our case, using fixed packet sizes results in $E[P] = E[P^*] = P$

Let T_b be the bit rate system throughput (expressed in bits/s), which is determined as

$$T_b = T_p \times E[P] \tag{6.3.18}$$

where $E[P]$ is average packet payload size (expressed as bits/packet).

6.4 Changes to Bianchi's model

Other authors have modified Bianchi's model to adapt it to better model real world network circumstances. Here follows a brief overview of the modifications that are applicable to this project.

6.4.1 Finite retry limit analysis

One of the first variations to Bianchi's model was to compensate for the finite retry limit (see: Figure 6.4.1) of the backoff process (see: [CPW02]). This model compensates for packet dropping when the maximum backoff stage is reached.

The random backoff timer, as described in section 6.3.2, is chosen in the interval $(0, CW-1)$, where CW is the contention window size. The size of CW depends on the number of collisions. At the first transmission attempt $W = CW_{min}$, which is the minimum contention window size. After each collision the contention window size doubles until it reaches the maximum value, $W_{m'} = CW_{max} = 2^{m'} CW_{min}$, where $W_{m'}$ is the largest contention window size. Once the maximum backoff timer value is reached it will remain there until it is reset. Therefore, we have

$$\begin{aligned} W_i &= 2^i \cdot W \quad & i \leq m' \\ W_i &= 2^{m'} \cdot W \quad & i > m' \end{aligned} \tag{6.4.1}$$

where i represents the backoff stage, i $(0, m)$, and m represents the short retry count and the maximum backoff stage value. With this modified model m can be larger than m', for example if $m = 7$ and $m' = 5$, with $CW_{min} = 32$, then $CW_{max} = 2^5(CW-1) = 1023$. Therefore if a packet reaches backoff stage 5 and experiences more collisions, it will thereafter choose a new backoff value randomly, in the range

$(0, 1023)$. The packet will only be discarded when it reaches backoff stage 7 and experiences yet another collision.

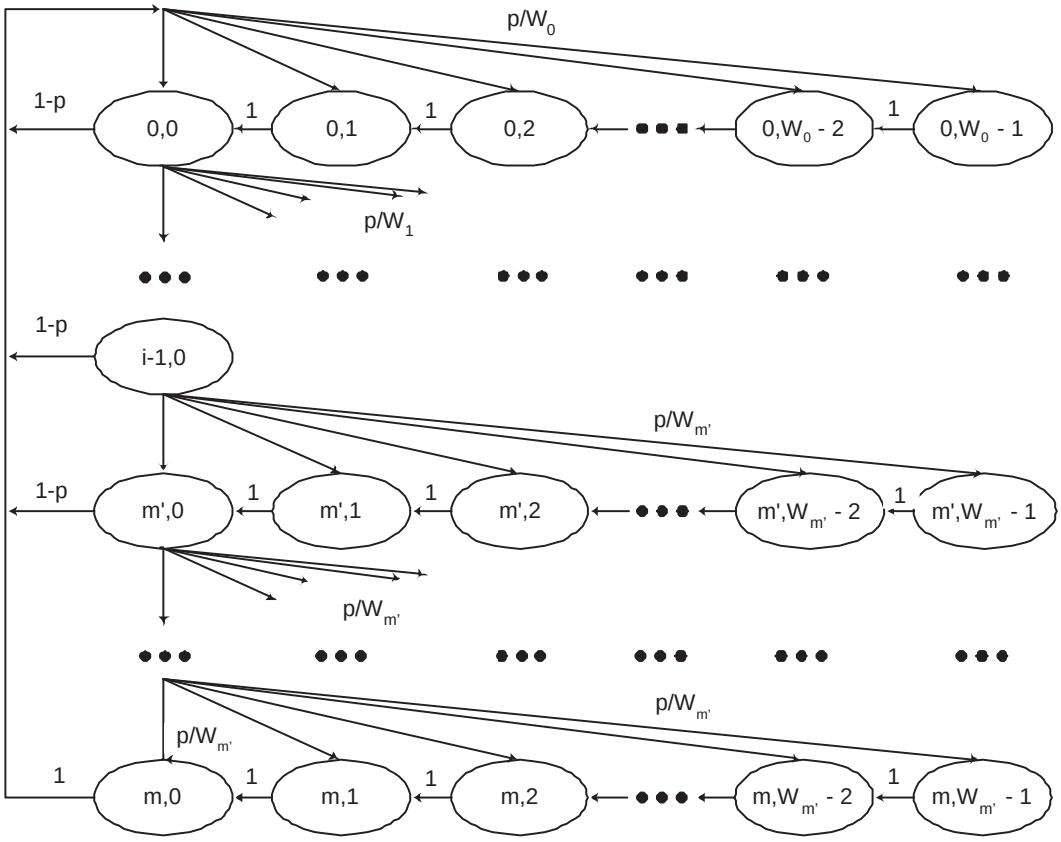


Figure 6.4: Bianchi's Markov chain with finite retry limit included

This approach better models the real 802.11 DCF protocol. It does, however, slightly change the mathematical analysis as follows: τ is now determined as

$$\tau = \sum_{i=0}^{m} b_{i,0} = \sum_{i=0}^{m} p^i \cdot b_{0,0} = b_{0,0} \cdot \frac{(1-p)^{m+1}}{(1-p)} \tag{6.4.2}$$

and

$$b_{0,0} = \begin{cases} \frac{2(1-2p)(1-p)}{(1-2p)(W+1)+pW(1-(2p)^m)} & ,m \leq m' \\ \frac{2(1-2p)(1-p)}{W\cdot(1-(2p)^{m'+1})\cdot(1-p)+(1-2p)\cdot(1-p^{m+1})+W\cdot 2^{m'}\cdot p^{m'+1}\cdot(1-2p)\cdot(1-p^{m-m'})} & ,m > m' \end{cases} \tag{6.4.3}$$

where p remains the conditional collision probability and W the maximum backoff value. The calculation for the probabilities of successful, idle or collision time slots, the throughput, and the time for successful and collision time slots otherwise remains the same as Bianchi's model.

6.5 Modeling a DCF non saturated network (Garetto's method)

First, the behaviour of a single station is studied by using a Markov model, to obtain the stationary transmission probability τ for any randomly chosen time slot by using a simplified model for the backoff process in the previous section [See: CG05]. Secondly, the implementation of empty buffer probabilities into the model will be described. Lastly, the entire model will be described.

6.5.1 Simplified model for saturated sources

This section describes a basic model for the behaviour of saturated sources, following the same approach as section 6.3 with a few minor simplifications.

6.5.1.1 Model assumptions

The main assumptions in [CG05], are identical to those indicated in Section 6.3.1. It is important to reiterate that the respective stations are all attempting to access the same wireless channel with an equal chance of access and with error-free communication.

6.5.1.2 Markov analysis for a simplified version of Bianchi's model

To develop an analytical model for a system such as this, a discrete Markov chain was used, and state changes at discrete time instants would be monitored. A fundamental assumption is made that the state of each station is independent of all others, reducing the analysis to monitoring only the behavior of a single tagged station. A typical depiction of the Markov chain under discussion is presented in Figure 6.5.

States annotated with b represent the scenario where the backoff timer has decremented to zero and the station attempts to transmits a packet in the current time step. The states that are annotated by B represent the scenarios where a station's backoff timer is still decrementing. As can be seen, this simplified version has combined all the backoff states into a single backoff state, which is B.

Each state has an index $0, 1, \ldots, m$ indicating the backoff stage, where m is the maximum retry limit for a single packet to be transmitted (if reached the packet is

discarded). Using the same notation as in [*Bia*00], W_i is the contention window size at backoff stage i. We have $W_i = min(2^i CW_{min}, CW_{max})$, where CW_{min} and CW_{max} are constant parameters representing the minimum and maximum backoff values.

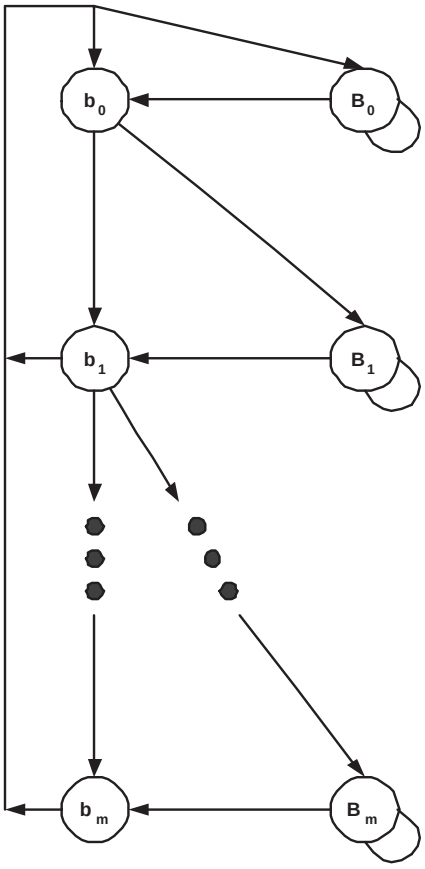


Figure 6.5: Bianchi's Markov chain with finite retry limit included

The transition probabilities from state s_i to s_f are $P(s_i, s_f)$, which is presented in Table 6.2, where i denotes the initial and f the final state. To improve the readability of the transition probabilities we have put $\alpha_i = 2/W_i$ and $\beta_i = 2/(W_i - 1)$. After transmitting a packet successfully or a collision, to avoid the unfair occurrence of a node picking a backoff value of zero and immediately transmitting again, a node has to wait at least one idle time slot before transmitting. Therefore, if a node picks a backoff value of 0 it is the same as picking a backoff value of 1. Note that including the factor of 2 in the numerator in α_i and β_i compensates for picking a backoff value of 0 or 1 which has the same effect as starting a transmission at the beginning of the next interval. This occurrence is also known as pre- and post-backoff, Figure 6.6 illustrates this.

The only unknown value to be determined from the transition probabilities is p,

Table 6.2: Summary of transition probabilities for simplified model

s_i	s_f	$P(s_i, s_f)$	Condition
b_i	b_0	$(1-p)\cdot\alpha_0$	$0 \leq i < m$
	B_0	$(1-p)\cdot(1-\alpha_0)$	
	b_{i+1}	$p\cdot\alpha_{i+1}$	
	B_{i+1}	$p\cdot(1-\alpha_{i+1})$	
b_i	b_0	α_0	$i = m$
	B_0	$1-\alpha_0$	
B_i	b_i	β_i	$0 \leq i < m$
	B_i	$1-\beta_i$	

which is the collision probability seen by a station transmitting on the channel. The stationary distribution of the Markov chain is denoted by $\pi = \{\pi_s\}$ where s is the generic state of the model. This is similar to Bianchi's model, where we have $b_i = p \cdot b_{i-1}$, for all $i > 0$.

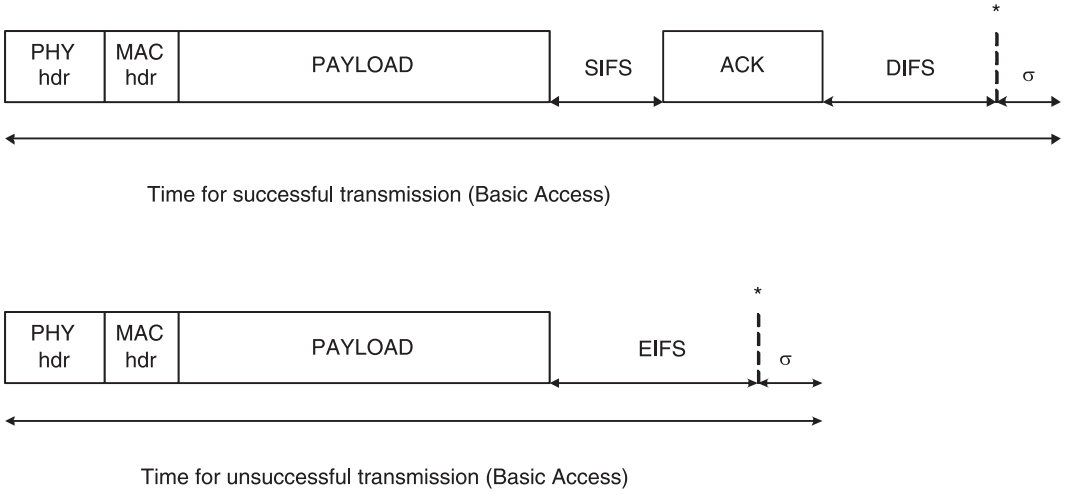


Figure 6.6: Bianchi's Markov chain with finite retry limit included

All states belonging to backoff stage i with a backoff counter value greater than one, have been collapsed into a single state B_i. This was done to reduce the number of states, which decreases the complexity of the model as it is extended to compensate for more variables. This approach leads to a marginal, but insignificant, compromise in accuracy, as will be shown in subsequent analysis.

The result is that the backoff is modelled as a geometrically distributed random variable instead of a uniformly distributed random variable. To obtain the same performance metrics as those of the collapsed version of the backoff values, with the more exact model having the uniformly distributed random variable, the following condition has to be satisfied,

$$B_i = b_i \frac{(W_i - 1)(W_i - 2)}{2W_i} \tag{6.5.1}$$

for all values of i. The reasoning behind the above scaling factor is that the transition probability from B_i to b_i is $2/(W_i - 1)$. This is to compensate for the post backoff mechanism. But from Bianchi's model we know that, using the same notation as in section 6.3.2, the state $b_{i,k} = ((W_i - k)/W_i) \cdot b_{i,0}$. Picking a value of 0 or 1 for this model, would be the same as picking a backoff counter value of 2 for Bianchi's model, so $b_{i,2} = ((W_i - 2)/W_i) \cdot b_{i,0}$. Therefore

$$\frac{2}{W_i - 1} B_i = \frac{(W_i - 2)}{W_i} b_i \quad \rightarrow \quad B_i = b_i \frac{(W_i - 1)(W_i - 2)}{2W_i} \tag{6.5.2}$$

Thanks to the particular structure of the Markov chain, all probabilities can easily be computed. All probabilities can be related to π_{b0}, and thus can be calculated by normalizing the overall sum of the probabilities to one, after which, one can calculate the probability τ that a station transmits in a given time slot as $\tau = \sum_{i=0}^{m} \pi_{bi}$. To compute the conditional collision probability p, the fundamental assumption used is that the individual stations are independent. With this assumption, it becomes possible to monitor the behavior of only a single station. This results in the following expression,

$$p = 1 - (1 - \tau)^{n-1} \tag{6.5.3}$$

The two unknowns to be determined are τ and p, which can be calculated by a simple iterative process. The probabilities that the channel is occupied with a successful transmission (Π_s), a collision (Π_σ), or an idle slot (Π_c), are respectively computed as follows:

$$\Pi_s = n\tau(1 - \tau)^{n-1} \tag{6.5.4}$$

$$\Pi_\sigma = (1 - \tau)^n \tag{6.5.5}$$

$$\Pi_c = 1 - \Pi_\sigma - \Pi_s \tag{6.5.6}$$

Finally, the aggregate packet throughput T_p (expressed in packets/s), is determined by

$$T_p = \frac{\Pi_s}{\Pi_s T_s + \Pi_c T_c + \Pi_\sigma \sigma} \tag{6.5.7}$$

where the denominator calculates the average duration of a time step.

6.5.2 Markov analysis for modelling a network with non saturated sources

In this section we describe the approach to deal with stations that have unsaturated sources. This implies that a station's queue might become empty, and thus that the analysis in section 6.3 would become invalid.

It is assumed that packets arrive from a higher layer at the MAC buffer, according to some external source with a rate of λ packets/s. The model is general enough to compensate for any traffic arrival process, but here we used a Poisson arrival process because of its simplicity and effectiveness. Also, if a packet arrives at an otherwise empty queue, a new backoff value is chosen. The result is a simpler model, especially at higher traffic loads. The Poisson packet arrival process is defined as

$$P_O[x] = \frac{(\Delta t \cdot \lambda)}{x!} e^{-(\Delta t \cdot \lambda)} \tag{6.5.8}$$

where λ is the average packet arrival rate,Δt the incremental time step and x the number of packet arrivals. $\Delta \cdot t$ is the duration of the time step and can be $\lambda \cdot T_s$ for a successful time slot, $\lambda \cdot T_c$ for a collision time step and $\lambda \cdot \sigma$ for an idle step.

The resulting model comprises of the states belonging to the set $\{b_{i,j,k}, B_{i,j,k}\}$, where b and B have the same meanings as in Section 6.3.2. The three indices $0 \leq i \leq m$, $0 \leq j \leq K$ and $0 \leq k \leq n$, respectively indicate the backoff stage (i), the number of packets currently stored in the buffer (j), and the active number of stations (k). Figures 6.5.2 and 6.5.2 depict an example of the Markov model. Figure 6.5.2 depicts only i and j, the use of k has been omitted. In addition, states $b_{0,0}$ and $B_{0,0}$ and the transitions from them are clearly illustrated. Figure 6.5.2 depicts only i and j, the use of k has been omitted. The working of the model is further illustrated here. Not depicted on this figure are multiple packet arrivals, which are not applicable to this project. The transition probabilities from state s_i to s_f are $P(s_i, s_f)$, where i denotes the initial and f the final state, of which some are presented in Table 6.3.

The MAC buffer (j) at each station is assumed to be of finite size, with a maximum value of K. If a new packet arrives and the MAC buffer has reached maximum capacity, the packet cannot be stored and will immediately be discarded. Further more, the extra index k indicates the number of active stations, keeping track of the number of stations having at least one packet in their queue.

The number k of active stations may fluctuate during a time step because of the following events: i) one or more of the stations having an empty buffer may receive new packets to be transmitted, increasing k; ii) a station that has only one packet successfully transmits, leaving the buffer empty, and decreases k. The possibility exists that these two events can occur simultaneously. Note, the 802.11 MAC protocol can only transmit a single packet successfully from one node during a time step, thus the number of active stations can only decrease by one, at most.

Table 6.3: Summary of transition probabilities for Garetto's model

s_i	s_f	$P(s_i, s_f)$	Condition
$b_{i,j,k}$	$B_{i,j+1,k+1}$	$(1-\alpha_0)q$	$i=0, j=0$
$B_{i,j,k}$	$B_{i,j+1,k+1}$	$(1-\alpha_0)q$	$i=0, j=0$
$b_{i,j,k}$	$B_{0,j-1,k}$	$(1-\alpha_0)(1-p)(1-q)$	$0 \leq i < m$,
	$b_{0,j-1,k}$	$(1-p)\alpha_0(1-q)$	$0 < j < k$,
	$B_{i+1,j,k}$	$(1-\alpha_0)(1-q)p$	$0 < k \leq N$
	$b_{i+1,j,k}$	$\alpha_0(1-q)p$	
	$b_{0,j,k}$	$(1-p)\alpha_0 q$	
	$B_{0,j,k}$	$(1-q)(1-\alpha_0)q$	
	$b_{i+1,j+1,k}$	$p\alpha_{i+1}q$	
$B_{i,j,k}$	$B_{i,j+1,k}$	$(1-\beta_i)q$	$0 \leq i < m$,
	$b_{i,j,k}$	$\beta_i(1-q)$	$0 < j < K$,
	$B_{i,j,k}$	$(1-\beta_i)(1-q)$	$0 < k \leq N$
	$b_{i,j+1,k}$	$\beta_i q$	
$b_{i,j,k}$	$b_{0,j-1,k}$	$\alpha_0(1-q)$	$i=m$,
	$B_{0,j-1,k}$	$(1-\alpha_0)(1-q)$	$0 \leq, j \leq K$,
	$B_{0,j,k}$	$(1-\alpha_0)q$	$0 < k \leq N$
	$b_{0,j,k}$	$\alpha_0 q$	

The number of stations joining the competing set depends on the duration of the time step Δt and the current number of competing stations k. During the interval Δt , the number of stations joining the competing set depends on the probability of q, where the probability of at least one packet arriving at the stations queue is $q = 1 - e^{-\lambda \Delta t}$. This probability is modeled according to a binomial distribution

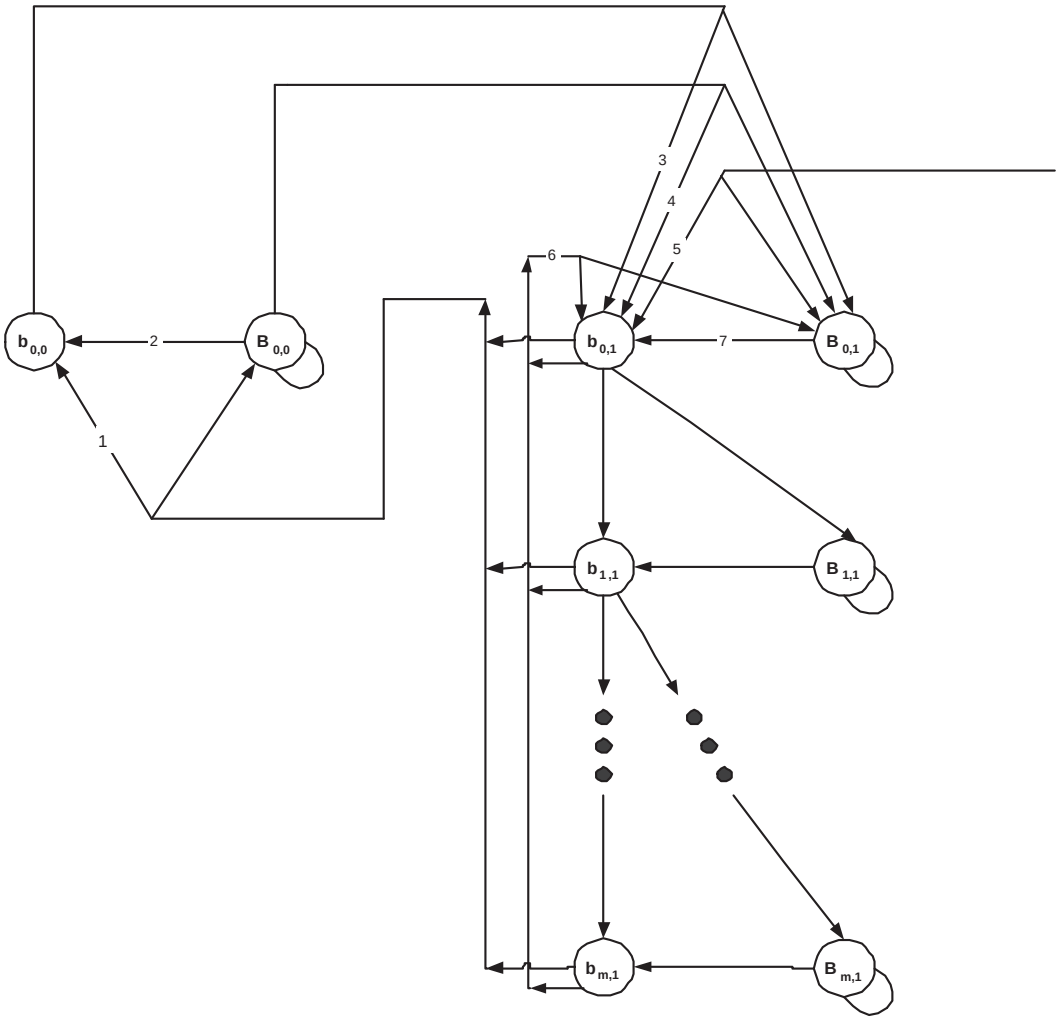


Figure 6.7: The Garetto Markov chain for states j=0 and j=1

with parameters (k,q,n), with k indicating the number of stations joining the competing set, q the packet arrival probability and n the number of non-active stations present.

$$B_N[k][n] = \binom{n}{k} q^k (1-q)^{n-k} \tag{6.5.9}$$

The possible time duration Δt directly relates to the probabilities that the channel is occupied by a successful (q_{good}), idle (q_{idle}) or collision (q_{bad}) time-slot. This implies, that in order to be able to calculate such probabilities, the probability $\tau(C)$ that a tagged station transmits in a given time step should be determined, where C indicates the number of competing stations, and $0 \leq C \leq n$ (n being the fixed maximum number of wireless stations present).$\tau(C)$ is determined in the same manner as in section 6.3.2, except more states are to be considered.

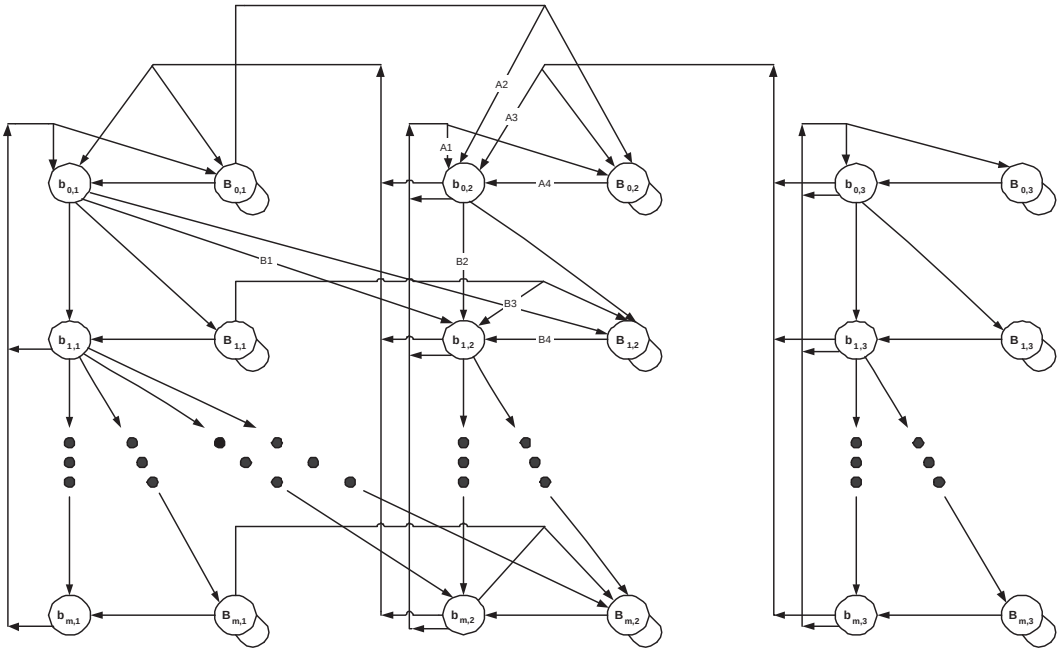


Figure 6.8: The Garetto Markov chain for stated j=1,2,3

The probability $\tau(C)$, $C \geq 1$, is given by:

$$\tau(C) = \frac{\sum_{i=0}^{m} \sum_{j=1}^{K} \pi_{b_{i,j,C-1}}}{\pi_{b_{0,0,C}} + \pi_{B_{0,0,C}} + \sum_{i=0}^{m} \sum_{j=1}^{K} (\pi_{b_{i,j,C-1}} + \pi_{B_{i,j,C-1}})} \tag{6.5.10}$$

where we have $C = k$ if $j = 0$, and $C = k+1$ if $j > 0$. If $C = 0$, we have $\tau(0) = 0$. Note that the probability $\tau(C)$ may vary from one state to another, depending on the value of C.

The numerator $\sum_{i=0}^{m} \sum_{j=1}^{K} \pi_{bi,j,C-1}$ compensates for all transmission states. The fact that $C = k+1$ if $j > 0$ compensates for when the tagged node does have a packet in its queue to transmit, increasing the number of competing stations by one. If $j = 0$, then there is one less competing station.

In the denominator $\sum_{i=0}^{m} \sum_{j=1}^{K} (\pi_{bi,j,C-1} + \pi_{Bi,j,C-1})$ accounts for all the states where there are one or more packets in the buffer. Again, if the tagged node does have a packet to transmit $j > 0$, the number of competing stations will include the tagged node as a competing station and $C = k+1$. $\pi_{b0,0,C} + \pi_{B0,0,C}$ adds the probability that there is no packet to transmit. The number of competing stations here is C, unlike the previous state probabilities where it was $C-1$. This compensates for the case where there is no packet to transmit $(j = 0)$.

A tagged station transmits in all states, given that $j > 1$. The conditional collision probability is given by

$$p(C) = 1 - [1 - \tau(C)]^{C-1} \tag{6.5.11}$$

Thus, with probability $1 - p(C)$, the transmission is successful. The probabilities $\Pi_s(C,k)$,$\Pi_c(C,k)$, and $\Pi_\sigma(C,k)$ respectively indicate that the channel is occupied with a successful transmission, a collision, or an idle time-slot. Further more, these probabilities are dependant on k and the total number of competing stations C:

$$\Pi_\sigma(C,k) = [1 - \tau(C)]^k \tag{6.5.12}$$

$$\Pi_s(C,k) = k\tau(C)[1 - \tau(C)]^{k-1} \tag{6.5.13}$$

$$\Pi_c(C,k) = 1 - \Pi_s(C,k) - \Pi_\sigma(C,k) \tag{6.5.14}$$

Remember that here k accounts for the number of competing stations excluding the tagged one, thus $0 \leq k \leq N-1$. Whereas $0 \leq C \leq N$ is used to indicate wether the tagged node does or does not have a packet to transmit, and determines whether it should be a competing station.

The last parameter of interest here is P_E, which is the probability that after a successful transmission of a packet, a station other than the tagged one finds its queue empty. This is a critical quantity to our model estimate, as we do not keep track of the buffer occupancy at other stations.

The approach used to calculate P_E is dependent on the number of competing stations C and the backoff stage i. A good approximation by [CG05] is whenever the tagged station finds itself at backoff stage i, and other stations are at backoff stage h, h differs from i by at most one, that is $|h-i| \leq 1$. The assumption behind this is the backoff stage of the tagged station will not significantly differ from that of the other competing stations. The estimate of P_E is defined as follows,

$$P_E(C,i) = \frac{\sum_{h:|h-i|\leq 1} \pi_{b_{h,1,C-1}}}{\sum_{h:|h-i|\leq 1} \sum_{j=1}^{K} \pi_{b_{h,j,C-1}}} \cdot e^{-\lambda T_s} \tag{6.5.15}$$

Note that at the beginning of the current time step there should be a packet available; there should also be no packet arrival. The probability that there is no packet arrival during a successful packet transmission is $e^{-\lambda T_s}$.

6.5.3 Typical example of transition probabilities

Table 6.4: Summary of transition probabilities

s_i	s_f	$P(s_i, s_f)$	Condition
$b_{i,j,k}$	$B_{i+1,j+1,k+1}$	$(1-\alpha_i)\cdot B_N[1][n]\cdot\Pi\cdot P_O[1]\cdot p$	$0\le i<m$
$b_{i,j,k}$	$B_{i+1,j+1,k+x}$	$(1-\alpha_i)\cdot B_N[x][n]\cdot\Pi\cdot P_O[1]\cdot p$	$0<j\le K$
$b_{i,j,k}$	$B_{i+1,j+x,k}$	$(1-\alpha_i)\cdot B_N[0][n]\cdot\Pi\cdot P_O[x]\cdot p$	$0<k\le N$
$b_{i,j,k}$	$b_{i+1,j,k}$	$(\alpha_i)\cdot B_N[0][n]\cdot\Pi\cdot P_O[0]\cdot p$	$0\le i<m$
$b_{i,j,k}$	$b_{0,j,k}$	$(\alpha_i)\cdot B_N[1][n]\cdot\Pi\cdot P_O[0]\cdot(1-p)$	$0<j\le K$
$b_{i,j,k}$	$b_{0,j,k}$	$(\alpha_i)\cdot B_N[0][n]\cdot\Pi\cdot P_O[1]\cdot(1-p)$	$0<k\le N$
$b_{i,j,k}$	$b_{0,j-1,k-1}$	$(\alpha_i)\cdot B_N[0][n]\cdot\Pi\cdot P_O[0]\cdot(1-p)$	
$B_{i,j,k}$	$B_{i,j,k-1}$	$(1-\beta_i)\cdot B_N[0][n]\cdot\Pi\cdot P_O[0]\cdot P_E[l][k]$	$\lvert l-i\rvert\le 1$ $0<j\le K$ $2\le k\le N$
$B_{i,j,k}$	$B_{i,j+1,k}$	$(1-\beta_i)\cdot B_N[0][n]\cdot\Pi\cdot P_O[1]$	$0\le i<m$
$B_{i,j,k}$	$b_{i,j,k}$	$\beta_i\cdot B_N[0][n]\cdot\Pi\cdot P_O[0]$	$0<j<K$ $1\le k\le N$
$b_{i,j,k}$	$b_{0,j-1,k}$	$(1-\alpha_0)\cdot B_N[0][n]\cdot\Pi\cdot P_O[0]$	$0\le i=m$ $1\le j\le K$ $1\le k\le N$

Table 6.4 illustrates typical transition probabilities. Not all transitions are included, but it serves as an example from which all transitions can be deduced.

Π represents the probability of a successful time slot, a collision or an idle time slot. These subscripts are omitted in the table for briefness.

B_N, P_O, and Π are all dependant on the action modelled, which can be a collision, idle or a successful transmission.

α_i and β_i have the same meaning as they did in section 6.5.1.2.

For example the first entry in the table, the transition probability from state $b_{i,j,k}$ to $B_{i+1,j+1,k+1}$ implies there was a collision, a new packet arrival and a new node other than the tagged one joining the competing set in the specific time step. The probability of a new station joining the competing step out of n inactive stations is $B_N[1][n]$, the probability of a single new packet arrival $P_O[1]$ and Π is the probability of a finding a collision in the current time step.

Another example would be from state $B_{i,j,k}$ to $B_{i,j,k-1}$, where the tagged node is and remains in backoff, and a node other than the tagged one now has an empty queue. So the transition probability from the current state to the next is $(1-\beta_i) \cdot B_N[0][n] \cdot \Pi \cdot P_O[0] \cdot P_E[l][k]$

Where the probability of no new nodes joining the competing set is $B_N[0][n]$, Π the probability of a good time slot, $P_O[0]$ is the probability of no new packets arriving at the tagged nodes packet queue, and $(1-\beta_i)$ the probability of remaining in backoff.

6.5.4 Numerical approach used

The numerical methods approach used to solve Garetto's model was a variation of the Bisection method [FM99]. This theorem states if a function $f(x)$ has a root and there are two points a and b, then if $f(a)$ and $f(b)$ have opposite signs, the root must lie between these two points. Making this interval smaller by dividing the distance between them in half will eventually lead to a result that converges.

The method used was to take the difference between the future state $(fs(s))$ and the current state $(cs(s))$, $\Delta(s) = fs(s) - cs(s)$. $\Delta(s)$ is a function of s, where s denotes the state variables $\pi_{bi,j,C-1}$ and $\pi_{Bi,j,C-1}$ for a specific backoff stage i, buffer occupancy j and k the number of competing stations. As we do not know the end point for this function, we give it an initial starting point. Depending on wether the value of $\Delta(s)$ is positive or negative the iterative process is as follows: If the sign is positive $fs(s)$ becomes

$$fs(s) = cs(s) + \Delta(s) \cdot 0.5 \cdot dir(s) \tag{6.5.16}$$

and if the sign is negative

$$fs(s) = cs(s) - \Delta(s) \cdot 0.5 \cdot dir(s) \tag{6.5.17}$$

$dir(s)$ is a variable added to speed up the iterative process by remembering the previous signs $\Delta(s)$ had. If there is no sign change implying that the sign has been the same every time the value is increased by one. For example the last ten signs have been positive, then $dir(s)$ would equal ten. If the sign does change, $dir(s)$ returns to the value of one.

For the model to reach steady state, an initial state is specified which is $\pi_{bi,j,k} = \pi_{b0,0,0} = 1$ and all other state probabilities are zero. This initial state is the system

starting from rest, no stations are active. Although this model would use a three dimensional matrix, two dimensional matrixes are used here to illustrate the principle. The transition probabilities are inserted into what is called the transition probability matrix, or transition matrix for short, as follows:

$$\widetilde{P} = p_{i,j,k} \tag{6.5.18}$$

where i, j, k respectively are still the backoff stage, number of packets, and number of competing stations.

A transition matrix whose transition probabilities sum to one and in which all entries are positive or zero, is stochastic. A Markov chain can be completely characterised by this one step transition matrix together with the initial state vector (see [Hok-97]).

Similarly, the state probabilities at each time interval can be expressed as a row vector.

$$\widetilde{\pi}^{(k)} = (\pi_0^{(k)}, \pi_1^{(k)}, ..., \pi_n^{(k)}) \tag{6.5.19}$$

Using matrix notation,

$$\begin{aligned} \widetilde{\pi}^{(1)} &= \widetilde{\pi}^{(0)}\widetilde{P} \\ \widetilde{\pi}^{(2)} &= \widetilde{\pi}^{(1)}\widetilde{P} \\ &\vdots \\ \widetilde{\pi}^{(k)} &= \widetilde{\pi}^{(k-1)}\widetilde{P} \end{aligned} \tag{6.5.20}$$

$$\widetilde{\pi}^{(k)} = \widetilde{\pi}^{(0)}\widetilde{P}^{(k)} \tag{6.5.21}$$

By back substituting $\widetilde{\pi}^{(i)}$ we get the following equation.

$$\widetilde{\pi}^{(k)} = \widetilde{\pi}^{(0)}\widetilde{P}^{(k)} \tag{6.5.22}$$

where $\widetilde{P}^{(k)}$ is the k-step transition matrix, that is the transition matrix $\widetilde{P}$ multiplied by itself k times.

A discrete Markov chain that is irreducible, aperiodic, and time homogeneous is said to be ergodic. For an ergodic Markov chain in matrix notation, the limiting probabilities or steady state values are

$$\widetilde{\pi} = \lim_{k \to \infty} \pi^{(k)} \tag{6.5.23}$$

The stationary probabilities are unique and determined through the following equations

$$\widetilde{\pi} \cdot \widetilde{e} = 1 \quad and \quad \widetilde{\pi} \cdot \widetilde{P} \tag{6.5.24}$$

where $\widetilde{e}$ is a row vector with all entries equal to one and this coincides with the normalization part of Garetto's model.

Also, to determine when the system reaches steady state, the relative error principle is used. The relative error is $R_p = \left| \frac{p - \hat{p}}{p} \right|$, where $\hat{p}$ is an approximation to p provided $p \neq 0$. The average number of active connections was used to determine the relative error. It is computed as

$$CON_{dist}[k] = \sum_{i=0}^{m} \sum_{j=0}^{N-1} \left(\pi_{b_{i,j,k}} + \pi_{B_{i,j,k}} \right) \qquad k \in [0, N-1] \tag{6.5.25}$$

$$E[con] = \sum_{k=0}^{N-1} k \cdot CON_{dist}[k] \tag{6.5.26}$$

With every iteration the relative error of the average number of connections is determined, where $\hat{p}$ will be the current value and p the value of the previous time instant. If the relative error is smaller than a predefined precision value, then the system has reached steady state. The precision value we used was 10^{-6}.

6.5.5 Experimental setup

The experimental setup is the same as that of section 7.1. The only two differences are the physical positioning of the nodes, which is not relevant as only the value specified for the propagation delay and the statistical approach matters here.

6.5.6 DCF mathematical model parameters

The same parameters as Chapter 5 are used. The DCF mathematical model was programmed in $Dev-C++$.

6.6 DCF Throughput results for the mathematical model

All definitions and factors on which the mathematical model's results will be dependant are as outlined in Chapter 5.

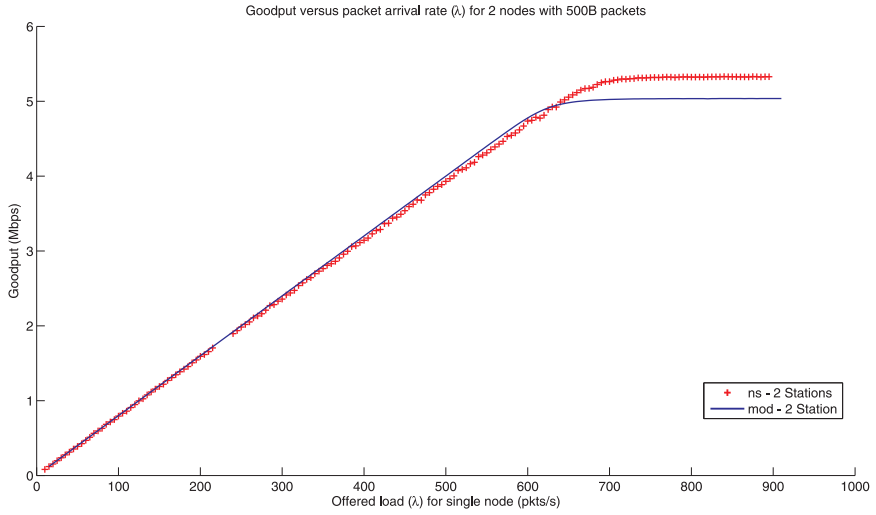


Figure 6.9: Goodput vs packet arrival rate comparison of mathematical modelling and simulation for 2 nodes with a constant 500B packet size

Table 6.5: Results for variable number of nodes for 500 B packets.

Number of nodes	Average error (kbps)	Percentage error	Average saturation throughput simulation (Mbps)	Average saturation throughput model (Mbps)
2	28.405	1.38	5.326	5.035
3	34.55	1.125	5.329	5.207
4	34.17	0.911	5.306	5.248
5	22.625	0.566	5.274	5.241
10	30.821	0.6641	5.064	5.103
15	61.395	1.2	4.97	4.896
20	96.774	2.081	4.852	4.775

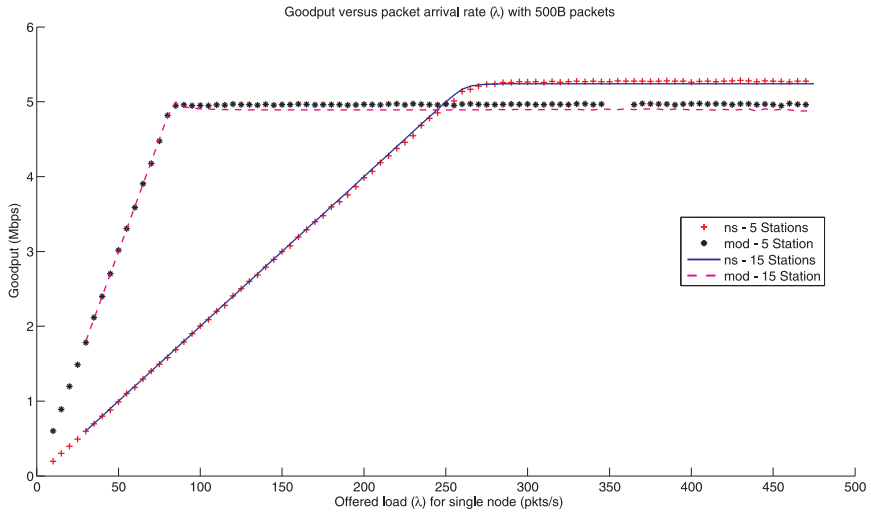


Figure 6.10: Goodput vs packet arrival rate comparison of mathematical modelling and simulation with a constant 500B packet size

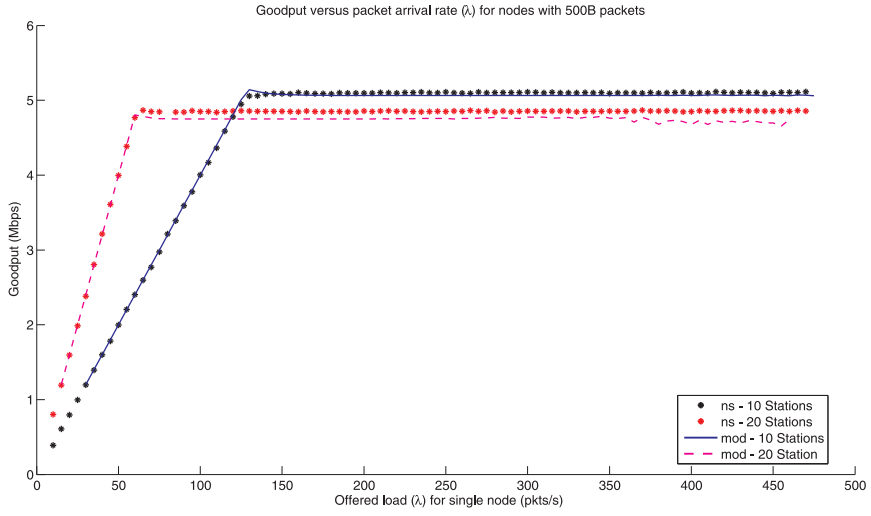


Figure 6.11: Goodput vs packet arrival rate comparison of mathematical modelling and simulation a constant 500B packet size

Table 6.6: Results for variable number of nodes for 1000 B packets.

Number of nodes	Average error (kbps)	Percentage error	Average saturation throughput simulation (Mbps)	Average saturation throughput model(Mbps)
2	201.68	3.8009	6.974	6.647
3	143.97	2.4583	6.919	6.727
4	120.31	1.9749	6.841	6.701
5	94.998	1.6808	6.77	6.642
10	157.15	2.59	6.479	6.31
15	261.17	4.2694	6.281	6.056
20	328.41	5.4543	6.139	5.863

From the results, a smaller number of stations does not model the unsaturated region as well as a larger number, which is attributed to the approximation of the probability of finding a queue, other than the tagged node, empty. This described in section 6.5.2 by equation P_E. It was shown in [CG05] that the approximation to P_E overestimates the throughput in the unsaturated region.

The conditional collision probability is determined as $p = 1-(1-\tau)^{n-1}$, which is dependent on τ and the number of nodes n. The error with which τ was determined has a less significant effect on p because of the $(1-\tau)$ term which is raised to the power $n-1$. If $(1-\tau)$ is a small value, determining it with a larger n value will have a small effect. However, the results contradict this, as the nodes are further increased from 10 nodes for the 500B and from 5 nodes for the 1000B cases respectively. As the number of stations increases, the number of states also increases, for example 20 stations with a buffer capacity of 20 and maximum retry limit of 7 have 5640 possible states. Note that with the same parameters except for 5 nodes, the number of states is 1610. More states require an increased number of iterations before the solution is obtained. With every iteration, if there is an error, the error in the final result increases. The more states, the greater the probability that there will be some error in computing them(refer to figure 6.12 in section 6.6.1.6 for a illustration of the overestimation of the conditional collision probability).

The results proves this to be a fairly accurate estimate for the throughput of DCF, with a maximum error of roughly 5 percent for 20 or fewer nodes.

6.6.1 DCF mathematical model for packet delay

In this section two mathematical models are identified to determine the queuing delay that would be compatible with Garetto's model. In the first part each model

will be described and in the second, the results. Lastly, the results will be compared with the results obtained by simulation in ns2.

6.6.1.1 Overview

A simple model was developed by Chatzimisios [BCV-03] for calculating the average packet delay E[D]. It was developed to be used with a model that is the slightly modified version of Bianchi's model in section 6.3; and takes into consideration the finite retry limit that drops packets. The results obtained from the Garetto model will be used to calculate the queueing delay. First the average packet delay will be discussed, and then the results from this mathematical model will be compared with that of the ns2 simulations.

6.6.1.2 Average packet delay (Chatzimisions method)

The average packet delay is defined as the time taken from when the packet is fetched from the queue by the MAC until the ACK is received, indicating the packet transmission was successful. If a packet was dropped because of reaching the maximum backoff stage value, the packet will not be included in the average packet delay calculation.

Provided the packet is not discarded, the average packet delay is defined as

$$E_C[D] = \sum_{i=0}^{m} (E[X_i] \cdot k_i) \tag{6.6.1}$$

where X_i is the average packet delay due to backoff at stage i, and k_i is the probability of the packet successfully reaching backoff stage i. $E[X_i]$ is given by

$$E[X_i] = d_i \cdot E[slot] \qquad , i \in [0,m] \tag{6.6.2}$$

where d_i is the average number of slot times the backoff counter decrements at stage i, and $E[slot]$ is the average slot time. d_i is given by

$$d_i = \sum_{k=0}^{W_i-1} \frac{W_i - k}{W_i} = \frac{W_i + 1}{2} \qquad , i \in [0,m] \tag{6.6.3}$$

The average slot time (E[slot]) has the same definition as that of Bianchi's and Garetto's models, which is

$$E[slot] = \Pi_s T_s + \Pi_c T_c + \Pi_\sigma \sigma \tag{6.6.4}$$

The probability k_i is given by:

$$k_i = \frac{p^i - p^{m+1}}{1 - p^{m+1}} \qquad , i \in [0,m] \tag{6.6.5}$$

where $1-p^{m+1}$ is the probability of a packet not being discarded. Equation 6.6.5 is the probability of a packet, if it is not dropped, reaching backoff stage i.

Combining equations 6.6.3 and 6.6.5 gives:

$$E[X] = \sum_{i=0}^{m} E[X_i] = \sum_{i=0}^{m} \left[\frac{W_i + 1}{2} \cdot \frac{(p^i - p^{m+1})}{1 - p^{m+1}} \right] \tag{6.6.6}$$

where $E[X]$ is the average number of time slots necessary for the successful transmission of any packet.

Finally, combining equations 6.6.6, 6.6.4 gives:

$$E_C[D] = E[slot] \cdot \sum_{i=0}^{m} \left[\frac{W_i + 1}{2} \cdot \frac{(p^i - p^{m+1})}{1 - p^{m+1}} \right] \tag{6.6.7}$$

6.6.1.3 Average packet delay (Vukovic method)

Another analytical model is that of Vukovic [see: SV-04] in which the average packet delay is calculated by the average delay for successfully received packets, after a specific number of collisions and the probability of these collisions. The average delay time of a packet is defined as the sum of the delays due to backoff, collisions and eventually the successful transmission. The average packet delay is given by

$$E_V[D] = \sum_{j=0}^{m} (E[V_j] \cdot q_j) \tag{6.6.8}$$

where V_j is the delay in the jth stage for a successfully transmitted packet and q_j the probability that a packet will be successfully sent from the jth stage.

The probability q_j of finding a node at stage j is given by:

$$q_j = \frac{p^j}{1 - p^{m+1}} \cdot (1-p) \qquad , j \in [0, m] \tag{6.6.9}$$

where p_j is the probability of the node being at stage j given that the packet is not dropped given by $1-p^{m+1}$. (1-p) is the probability the packet is successfully transmitted. The average delay for stage j is given by:

$$E[V_j] = T_s + j \cdot T_c + E[slot] \cdot \sum_{i=0}^{j} \left[\frac{W_i - 1}{2} \right] \qquad , j \in [0, m] \tag{6.6.10}$$

where T_s and T_c are the times for a collision and successful packet transmission, respectively. $(Wi-1)/2$ is the average number of backoff time slots, jT_c is the time occupied because of j collisions and $E[slot]$ is the average time slot as defined in the previous section.

Finally we get Vukovic's delay as

$$E_V[D] = \sum_{j=0}^{m} \left(\left(T_s + j \cdot T_c + E[slot] \cdot \sum_{i=0}^{j} \frac{W_i - 1}{2} \right) \cdot \frac{p^j \cdot (1-p)}{1 - p^{m+1}} \right) \tag{6.6.11}$$

6.6.1.4 Corrected Vukovic delay

Minor modifications were made to Vukovic's model in [ABCPRV05]. The only change is the way in which the average slot time is determined. The changes will be summarised here. The modified delay is now determined as

$$E_{VN}[D] = \sum_{j=0}^{m} (E[VN_j] \cdot q_j) \tag{6.6.12}$$

where VN_j is the average delay in the jth backoff stage and q_j the probability the packet will be transmitted. q_j is determined as in the previous section by equation 6.6.9.

The average delay $E_{VN}[D]$ is calculated by

$$E_{VN}[j] = T_s + j \cdot T_c + E'[slot] \cdot \sum_{i=0}^{j} \left[\frac{W_i - 1}{2} \right] \qquad , j \in [0, m] \tag{6.6.13}$$

$E'[slot]$ is now determined by

$$E'[slot] = (1 - P'_{tr}) \cdot \sigma + P'_{tr} \cdot P'_s \cdot T_s + P'_{tr} \cdot (1 - P'_s) \cdot T_c \tag{6.6.14}$$

where P'_{tr} is the probability that at least one out of the $n-1$ stations transmits during a time slot. It is given by

$$P'_{tr} = 1 - (1 - \tau)^{n-1} \tag{6.6.15}$$

P'_s is the probability of successful packet transmission occurring given that only one of the $n-1$ remaining stations transmits a packet. P'_s is given by

$$P'_s = \frac{(n-1) \cdot \tau \cdot (1-\tau)^{n-2}}{P'_{tr}} = \frac{(n-1) \cdot \tau \cdot (1-\tau)^{n-2}}{1 - (1-\tau)^{n-1}} \tag{6.6.16}$$

Finally $E_{VN}[D]$ is as follows:

$$E_{VN}[D] = \sum_{j=0}^{m} \left(\left(T_s + j \cdot T_c + E'[slot] \cdot \sum_{i=0}^{j} \frac{W_i - 1}{2} \right) \cdot \frac{p^j \cdot (1-p)}{1 - p^{m+1}} \right) \tag{6.6.17}$$

The reasoning behind these changes is when the tagged node is deferring, it cannot contend for the channel. Vukovic did not take this into consideration as his model's average slot time ($E[slot]$) was determined in actuality for $n+1$ stations.

6.6.1.5 Empty queue probability

From Garetto's model the average empty queue probability had to be determined in order obtain accurate queueing delay results. The connection distribution ($CON_{dist[k]}$) is determined as

$$CON_{dist}[k] = \sum_{i=0}^{m} \sum_{j=0}^{N-1} \left(\pi_{b_{i,j,k}} + \pi_{B_{i,j,k}} \right) \qquad k \in [0, N-1] \tag{6.6.18}$$

and the expected number of connections ($E[con]$) is determined as

$$E[con] = \sum_{k=0}^{N-1} k \cdot CON_{dist}[k] \tag{6.6.19}$$

The average number of connections is used to determine the empty queue probability by $E[empty] = (N - E[con])/N$. Thus multiplying $E[Empty]$ by $Ec[D]$ and adding this to the single packet delay gives

$$E[T] = E_c[D] + E[Empty] \cdot E_c[D] \tag{6.6.20}$$

6.6.1.6 Chatzimisios and Vukovic models comparison of results and discussion

In Figure 6.13 both the Chatzimisios and Vukovic methods are used to determine the delay of a packet measured from the time instant the packet is fetched from the queue until the successful reception of the ACK. This figure, clearly indicates the Chatzimisios method to be the better, and it is therefore the method used for

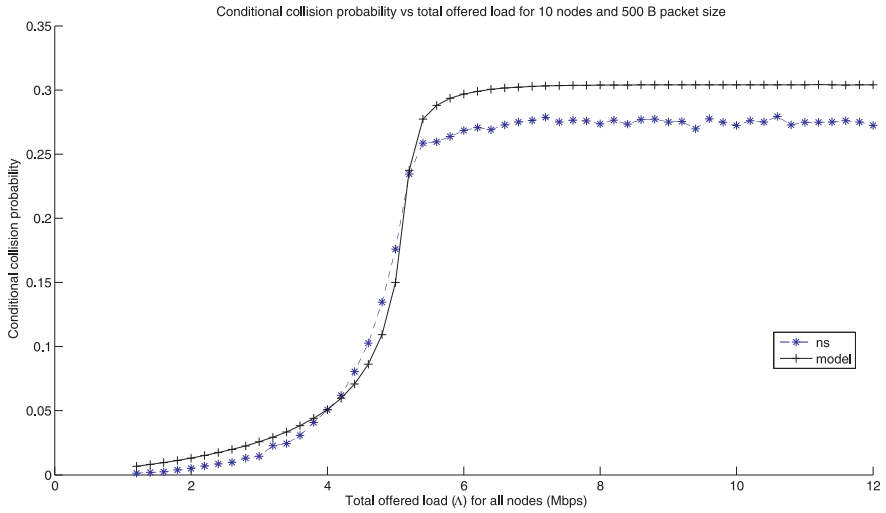


Figure 6.12: Conditional collision probability vs the total offered load for 10 nodes (500B packet size)

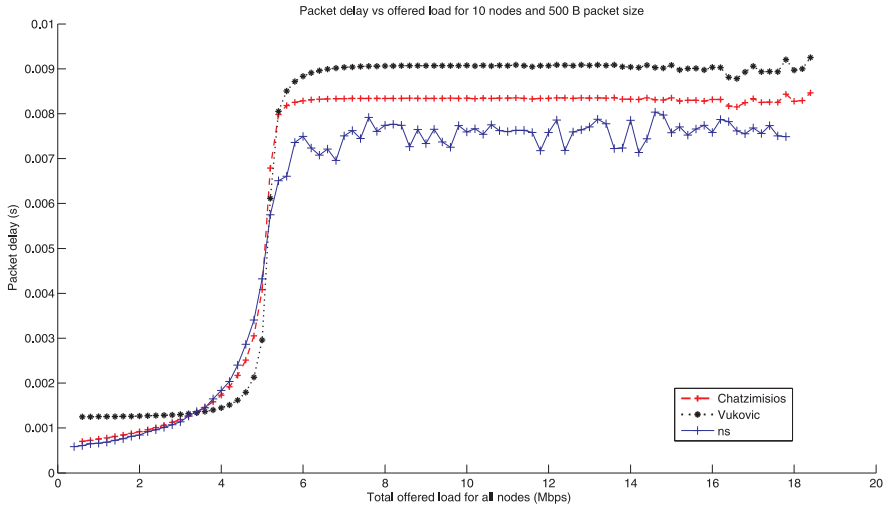


Figure 6.13: Packet delay vs the total offered load for the Chatzimisios and Vukovic methods for 10 nodes (500B packet size)

all other delay calculations from here onward. The reason this method is better is that the conditional collision probability computed by Garetto's model is a good approximation, but not perfect, as can be seen in Figure 6.12, for which the relative error is 8.8793%.

Because of this error both methods will estimate the packet delay with some error and although both require the same number of multiplication operations, Vukovic's

Table 6.7: Packet delay results for constant 500B size

Number of nodes	Average error (ms)	Percentage error (%)	Average saturation delay simulation (ms)	Average saturation delay model (ms)
2	0.11945	11.775	1.501	1.687
3	0.17126	10.753	2.242	2.447
4	0.20113	8.931	3.001	3.234
5	0.19294	7.9337	4.045	3.792
10	0.4306	6.7478	7.926	8.339
15	0.76518	7.1614	12.03	12.83
20	1.4797	9.7055	16.75	17.3

model increases this error by multiplying with the $T_s + j \cdot T_c + E[slot] \cdot \sum_{i=0}^{j} \frac{W_i - 1}{2}$ term, thus the error is additively increased. Also, not shown here but in [ABCPRV05] the authors illustrate that the fewer the number of nodes the larger the packet delay error becomes for Vukovic's model, compared to the Chatzimisios method.

6.6.2 Chatzimisios method: Results and discussion

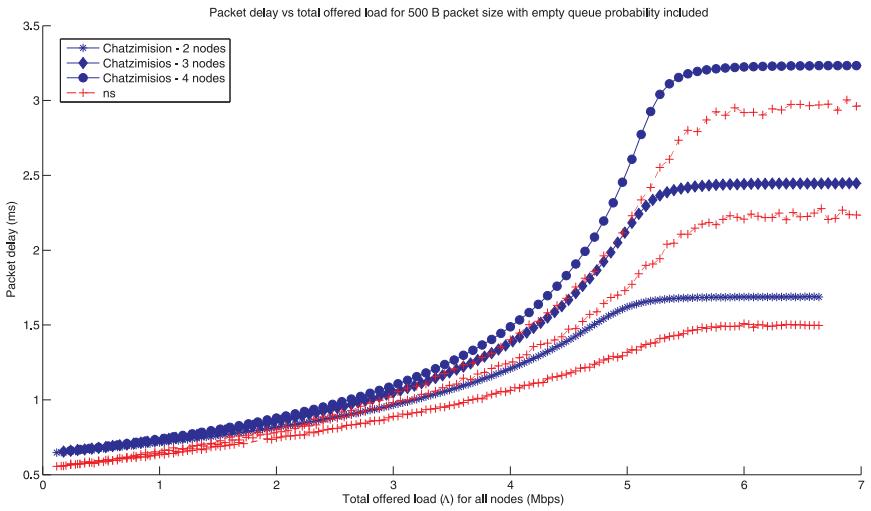


Figure 6.14: Packet delay vs total offered load comparison of mathematical modelling and simulation for 2,3 and 4 nodes (500B packets)

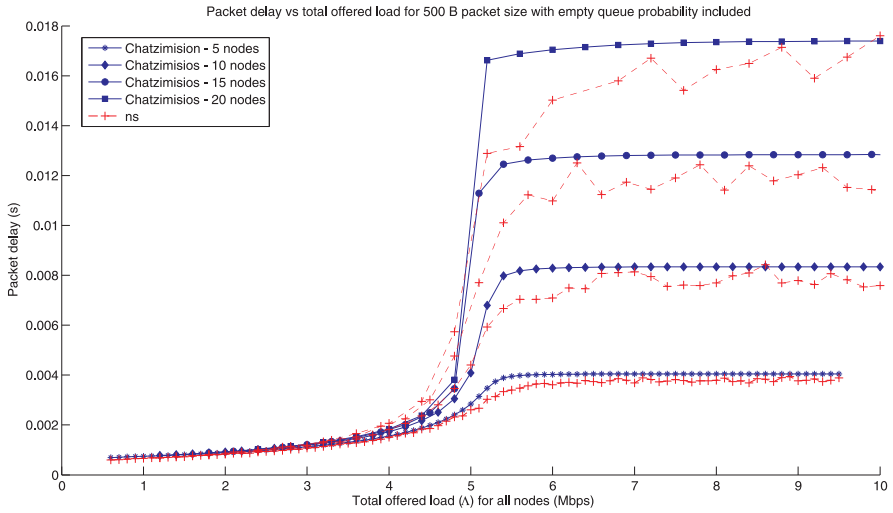


Figure 6.15: Packet delay vs total offered load comparison of mathematical modelling and simulation for 5,10,15 and 20 nodes (500B packets)

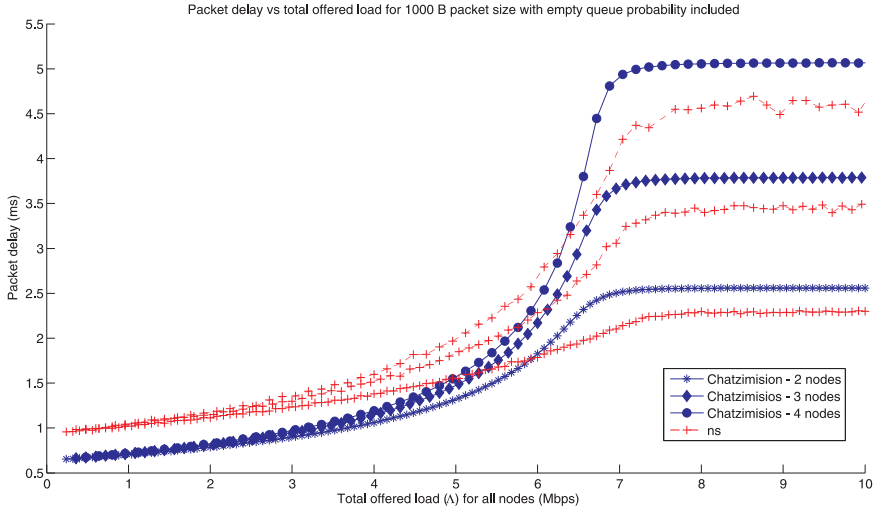


Figure 6.16: Packet delay vs total offered load comparison of mathematical modelling and simulation for 2,3 and 4 nodes (1000B packets)

From the Figures 6.14, 6.15, 6.15 and 6.16 it is clear the Chatzimisios method is an adequate approximation of the packet delay obtained from the ns2 simulations. In the unsaturated region especially for the lower number (for example 2, 3 and 4) nodes and both packet sizes 500B and 1000B, Garetto's model estimates the empty queue probability with slightly more inaccuracy, because the assumption that all nodes are at roughly the same backoff stage is less valid, due to rapid fluctuations of more packets being transmitted.

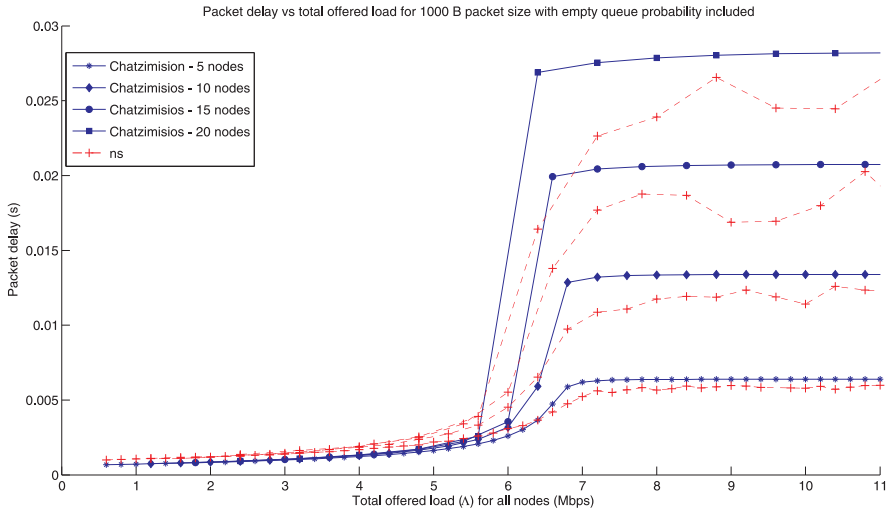


Figure 6.17: Packet delay vs total offered load comparison of mathematical modelling and simulation for 5,10,15 and 20 nodes (1000B packets)

Table 6.8: Packet delay results for constant 1000B size

Number of nodes	Average error (ms)	Percentage error (%)	Average saturation delay simulation (ms)	Average saturation delay model (ms)
2	0.27675	14.926	2.306	2.557
3	0.33526	11.699	3.473	3.787
4	0.41693	10.525	4.694	5.064
5	0.46399	12.187	5.913	6.385
10	1.1659	10.996	12.34	13.39
15	1.6636	10.072	18.96	20.74
20	2.5251	11.264	26.42	28.2

In the saturated region of the packet delay plot, the conditional collision probability, as was shown in Figure 6.12, does not have the same gradual changeover as that estimated by ns2. This leads to the mathematical model, when reaching, saturation to almost immediately reach the maximum value. The estimated value of the conditional probability is always slightly larger than that of ns2 and, therefore, the expected packet delay will always be larger in the saturated region. Although, as can be seen from the figures, the fluctuation from the simulated values in the saturation region of the maximum outliers is roughly the mathematical value.

In section 6.6.1.6 it was shown that the relative error for 10 nodes with a constant 500B packet size was 8.8793% for the conditional collision probability. From Table6.14 the relative error for the packet delay with the same parameters came to 6.7478%, which agrees with the conclusion that the error between the simulated and mathematical results is due to the overestimation of the conditional collision probability obtained by Garetto's model.

The results are in agreement with the results obtained in section 6.6, because a larger packet delay indicates a lower packet throughput. The overestimation of the packet delay implies that the mathematical model for the DCF throughput will underestimate the simulated throughput in the saturated region.

6.7 DCF buffer occupancy

In Section 6.6.1 the time taken to transmit a single packet was determined, and is defined as the interval of time from when the packet is removed from the head of the queue until the ACK is received for its successful reception. In this section it is necessary to determine the average buffer occupancy to aid in the calculation of the total queueing delay for a single packet, defined as the time interval from when the packet arrives in the queue until the reception of the ACK indicating its successful transmission.

Garetto's model was used to determine the average buffer occupancy, as the buffer occupancy is one of the state parameters modelled.

For an empty buffer the queue length probability is determined as

$$QL_{dist}[0] = \sum_{k=0}^{N-1} \left(\pi_{b_{0,0,k}} + \pi_{B_{0,0,k}} \right) \tag{6.7.1}$$

where k is the number of stations present. Notice the inclusion of the probability of the idle slot time and the probability of the idle slot.

A buffer queue with one or more packet ($j > 0$) queue length distribution is computed as follows,

$$QL_{dist}[j] = \sum_{i=0}^{m} \sum_{k=0}^{N-1} \left(\pi_{b_{i,j,k}} + \pi_{B_{i,j,k}} \right) \tag{6.7.2}$$

Then all the queue length probabilities are summed and used to normalize the queue length distribution

$$QL_{sum} = \sum_{j=0}^{K} QL_{dist}[j] \tag{6.7.3}$$

and

$$QL_{norm}[j] = \frac{QL_{dist}[j]}{QL_{sum}} \tag{6.7.4}$$

from where the average is determined for the expected buffer occupancy

$$E[buf] = \sum_{j}^{K} j \cdot QL_{norm}[j] \tag{6.7.5}$$

6.7.1 Buffer occupancy results and discussion

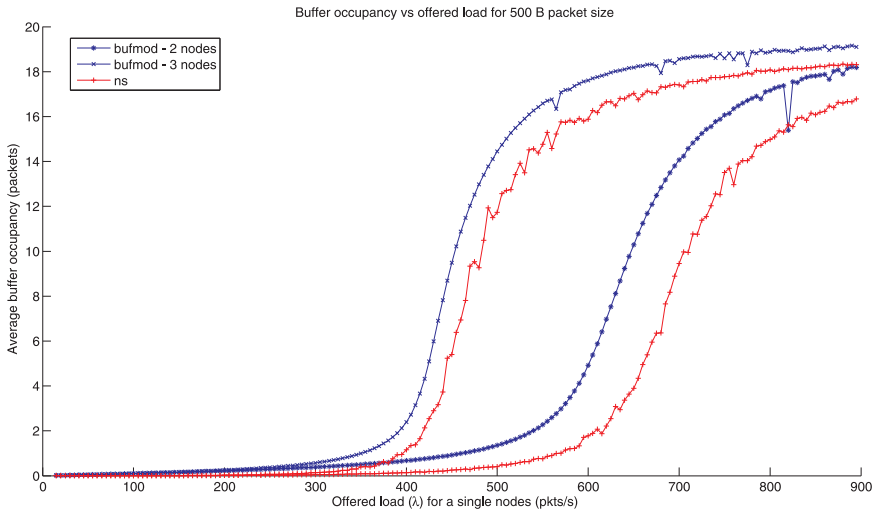


Figure 6.18: Buffer occupancy vs packet arrival rate comparison of mathematical modelling and simulation for 2 and 3 nodes (500B packets)

As seen from the figures (see figure 6.18 to 6.21) in this section, this method is fairly accurate. Also from tables 6.9 and 6.10 it will also be noted that the percentage error and average error decreases as the number of nodes increases. We will define the transition region as the buffer queue having one to seventeen packets, and as can be observed, this is the region contributing the most to the overall error. We define

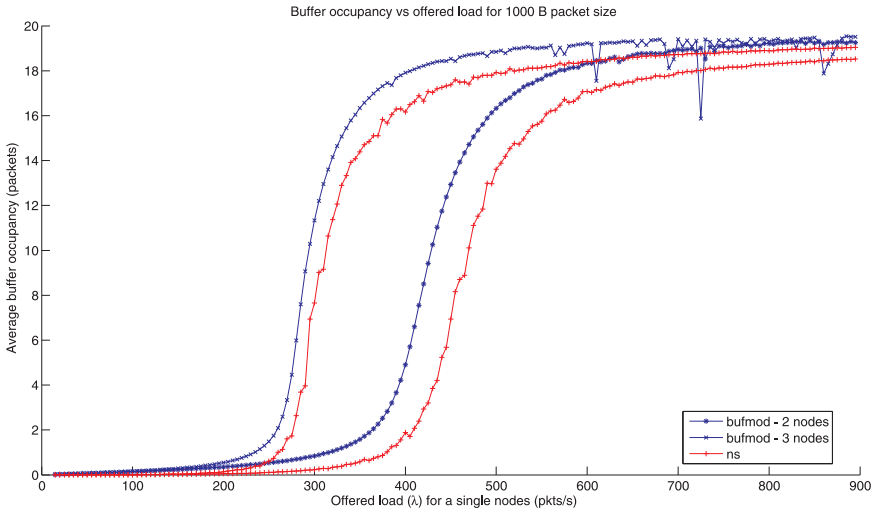


Figure 6.19: Buffer occupancy vs packet arrival rate comparison of mathematical modelling and simulation for 2 and 3 nodes (1000B packets)

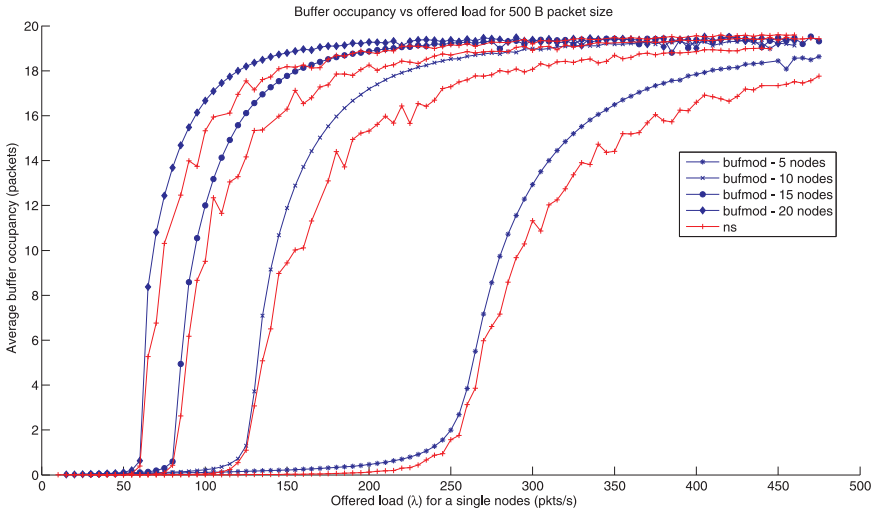


Figure 6.20: Buffer occupancy vs packet arrival rate comparison of mathematical modelling and simulation for 5,10,15 and 20 nodes (500B packets)

the saturation region as any buffer occupancy larger than seventeen packets. The value of seventeen is not the start of the saturation region, it is used for illustrative purposes, as well as the definition of the transition region.

It can therefore be noted that the buffer overestimation by the model is larger in the transition region than the saturated region. In the saturated region, the larger the packet arrival probability becomes, the more the mathematical and simulated results

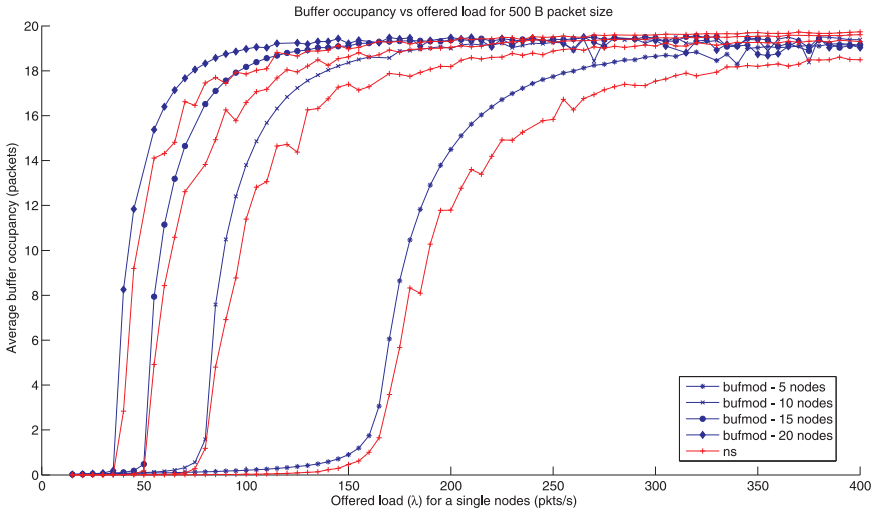


Figure 6.21: Buffer occupancy vs packet arrival rate comparison of mathematical modelling and simulation for 5,10,15 and 20 nodes (1000B packets)

Table 6.9: Average buffer occupancy results for variable number of nodes 500B constant packet size

Number of nodes	Average error (packets)	Percentage error (%)
2	1.135	28.722
3	1.0137	12.227
4	0.76255	7.132
5	0.85936	12.62
10	1.0229	8.5765
15	0.57702	3.7905
20	0.39869	2.4922

converge. Accordingly, the longer the duration of the transition region, the larger the error will be. For fewer nodes this region is longer. The error is also influenced by the population sample over which it is measured and the packet arrival rate.

The saturation region is better approximated by the mathematical model, which results in a smaller error. Therefore, if the sample range is the same for five and for twenty nodes, the error will be less for twenty nodes because a higher percentage of the range falls in the saturated region.

It was shown in [CG05] that the approximation for $P_E(C,i)$ used (refer to equation 6.5.15) is to blame for the overestimate of the buffer occupancy; another approach

Table 6.10: Average buffer occupancy results for variable number of nodes 1000B constant packet size

Number of nodes	Average error (packets)	Percentage error (%)
2	1.3188	15.118
3	0.88136	7.3196
4	0.69087	5.0217
5	0.84771	0.07968
10	0.59153	0.03779
15	0.55744	0.0326
20	0.58556	0.03338

was used to determine P_E and this again underestimated the buffer occupancy. The assumption is less valid for a lower number of competing nodes with backoff stages which differ by one. Fluctuations in the backoff stage are more rapid for fewer nodes because a single node transmits more packets. This approximation also overestimates the number of competing stations.

The results are in agreement with the results obtained in section 6.6 because larger average buffer occupancies indicate a lower packet throughput. The overestimation of the packet delay (section 6.6.1) and buffer occupancy implies that the mathematical model for the DCF throughput will underestimate the simulated throughput in the saturated region.

6.8 End-to-end delay

Little's law states that the average end-to-end delay for a packet can be determined by the product of the average packet delay (determined in section 6.6.1) and the average buffer occupancy (determined in section 6.7.1) as follows:

$$e2e\_delay = E[buf] \cdot E[T] \tag{6.8.1}$$

6.8.1 Results and discussion

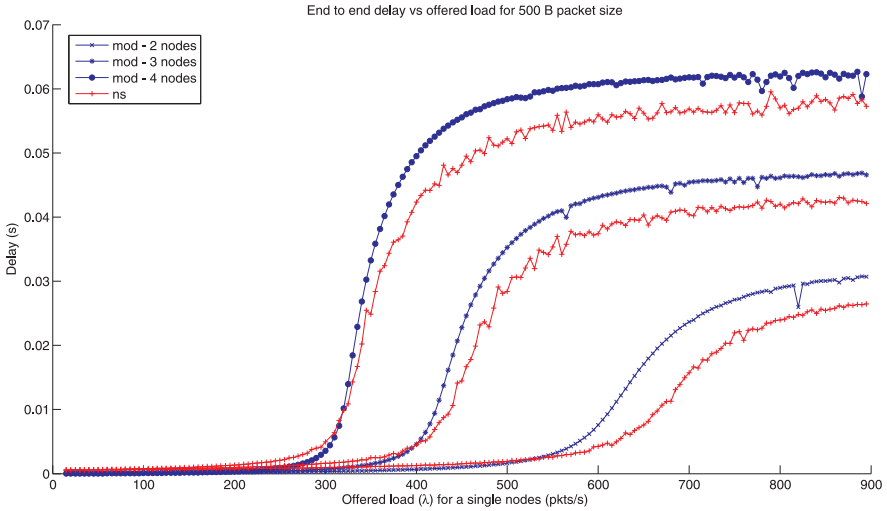


Figure 6.22: End-to-end delay vs packet arrival rate comparison of mathematical modelling and simulation for 2,3 and 4 nodes (constant 500B packet size)

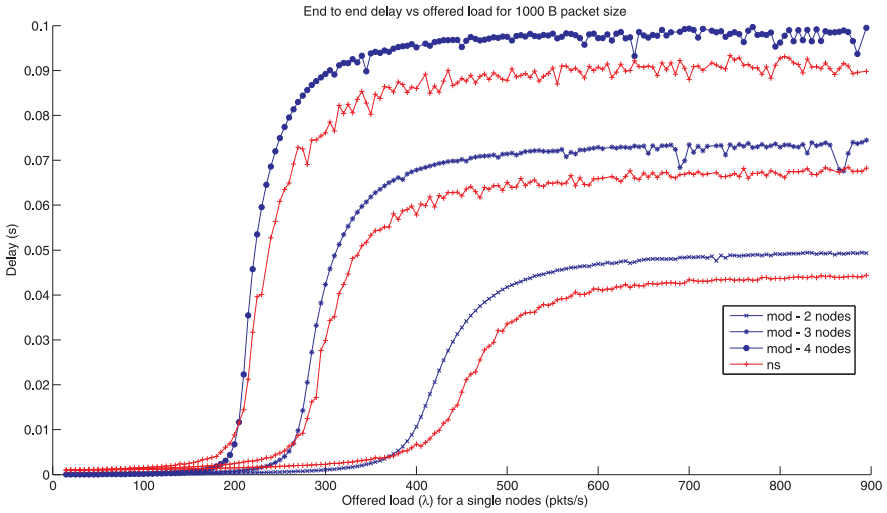


Figure 6.23: End-to-end delay vs packet arrival rate comparison of mathematical modelling and simulation for 2,3 and 4 nodes (constant 1000B packet size)

The results obtained (see figures 6.22 to 6.25, and tables 6.11 and 6.12) are in agreement with sections 6.6, 6.6.1 and 6.7.1. The average buffer occupancy and average packet delay are overestimated by the model, which implies that their multiplication

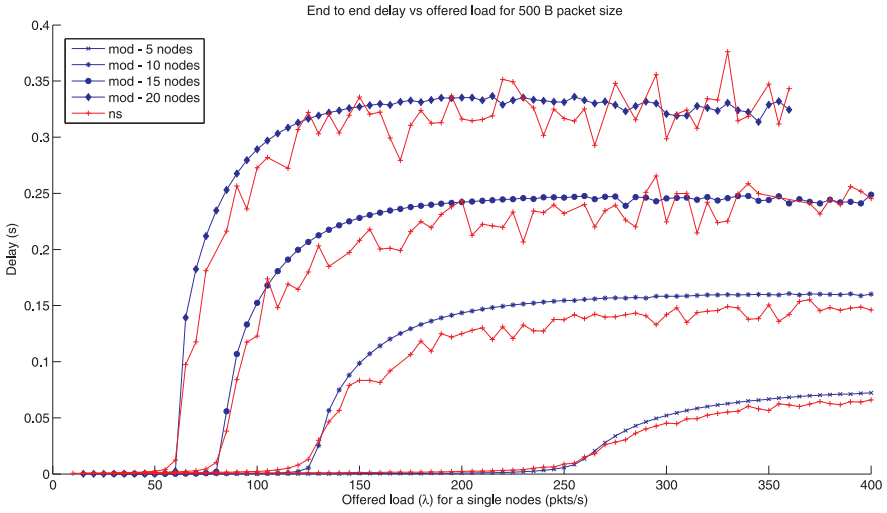


Figure 6.24: End-to-end delay vs packet arrival rate comparison of mathematical modelling and simulation for 5,10,15 and 20 nodes (constant 500B packet size)

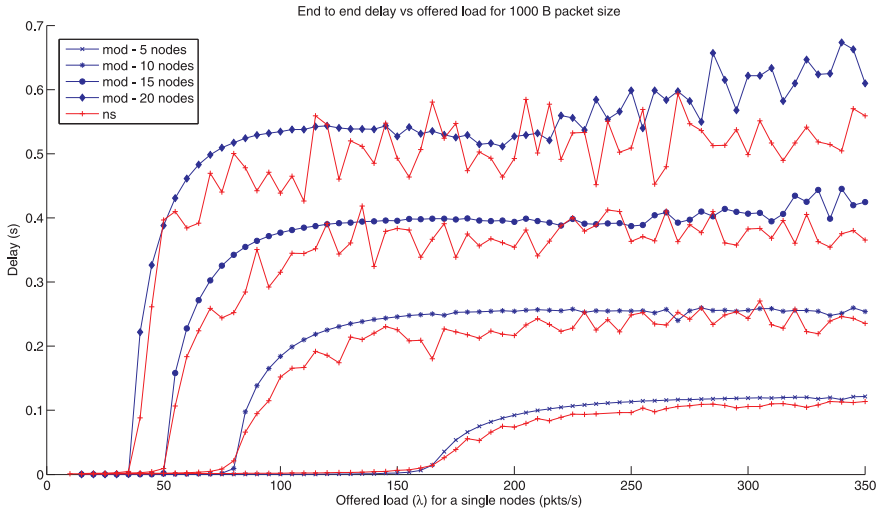


Figure 6.25: End-to-end delay vs packet arrival rate comparison of mathematical modelling and simulation for 5,10,15 and 20 nodes (constant 1000B packet size)

Table 6.11: End-to-end packet delay results for a variable number of nodes with constant 500B packet size

Number of nodes	Average error (ms)	Percentage error	Average saturation delay simulation (ms)	Average saturation delay model (ms)
2	1.9522	27.562	26.38	30.52
3	3.0201	15.037	42.62	46.05
4	3.6655	10.77	57.96	61.88
5	3.7581	13.424	69.24	75.19
10	1.318	13.9	152.4	160.5
15	1.4113	7.3764	239.4	247.5
20	1.5012	5.8439	325.2	332.4

Table 6.12: End-to-end packet delay results for a variable number of nodes with constant 1000B packet size

Number of nodes	Average error (ms)	Percentage error	Average saturation delay simulation (ms)	Average saturation delay model (ms)
2	4.3735	20.063	49.2	43.7
3	5.3416	12.303	73.02	67.22
4	6.9499	10.514	98.23	90.96
5	7.4869	11.177	121.9	112.4
10	22.7	11.589	255.8	241.2
15	29.484	9.8154	397.7	370.6
20	48.95	11.039	547.6	492.3

will also overestimate the end-to-end delay. Accordingly, the multiplication of the two errors will further increase the total error.

6.9 Summary

In this chapter a mathematical model to obtain the throughput, packet delay, buffer occupancy and end-to-end delay of a DCF 802.11 network was discussed. The results obtained were discussed and compared with those from ns2. It was found that the throughput was underestimated by the model, which resulted in the overestimation of the packet delay, buffer occupancy and end-to-end delay.

Chapter 7

PCF mathematical model

In this chapter a mathematical model will be described that was used to predict the behaviour of the PCF mode of operation in IEEE 802.11 networks. A mathematical approach will be described, determining first the throughput and then the packet delay. The packet delay for PCF is more involved than that for DCF because different approaches had to be used for saturated and unsaturated packet conditions.

7.1 Motivation for creating a mathematical model

To verify theoretically that our simulation results are correct, it is imperative to create a mathematical model which does this. The results gained from this will also aid in the decision making process as to which communication protocols are better suited to which network conditions.

7.2 Service cycle time

7.2.1 Definition

For this project it was decided that the Point Coordinator (PC) should adopt a pure round-robin polling scheme. The service cycle time is the time it takes the PC to service all nodes on the polling list and we determine the expected service cycle time as (see [Sikbar-05])

$$E[X_{SC}] = B + iV + i\rho L \tag{7.2.1}$$

where B is the time used which can be attributed to overhead incurred by beacon and CFEnd frames in every CFP, V the duration of every poll transmitted, i the number of pollable nodes, L the expected transmission time of a data packet and ρ the server utilization. This equation holds true if there is enough time for all nodes on the polling list to be serviced in one superframe.

The utilization for PCF is $\rho = \lambda T_{SF}$, where lambda is the packet arrival rate and T_{SF} the superframe time. Saturation, and the maximum service cycle time, is reached when every station has a packet to send during every CFP, for which the utilization would be at its maximum value of one.

The duration of each polling frame (V) is the time taken to transmit a single poll packet with the SIFS time included. The expected time to transmit a data packet (L) includes the time for a node to transmit a data packet, the time necessary to receive the ACK packet, and the SIFS waiting time before the following poll transmission.

A miner modification made to the formula for the purpose of this project was to compensate for the manner in which the PCF toolbox of *ns*2 handles pollable nodes which do not have data to send. If a station does not have data to send, the base station waits a PIFS time period before the next node on the polling list is polled. The new formula is

$$E[X_{SC}] = B + iV + i\rho L + i(1-\rho)T_{PIFS} \tag{7.2.2}$$

where $1-\rho$ is the probability of no data packet being available for transmission when a poll is received and T_{PIFS} the time for a PIFS time period.

7.3 PCF throughput

7.3.1 Mathematical model

A simple method to determine the expected throughput was devised for this project by doing the following:

$$E[Throughput] = i \cdot \lambda \cdot E[P] \tag{7.3.1}$$

where i is the number of pollable nodes, $E[P]$ the average packet size and λ the average packet arrival rate.

λ_{sat} is defined as the value at which the combined throughput of the pollable nodes saturates.

$$\lambda_{Sat} = \frac{1}{T_{SF}} \tag{7.3.2}$$

If λ in equation 7.3.1 is less than λ_{Sat} then the value for λ is used, and if the value of λ is larger than λ_{Sat} then λ_{Sat} is used.

7.3.2 Results and discussion

The results prove to be an accurate approach to determine the combined throughput for nodes using the Round Robin Polling method, with a relative error of less than one percent. The goodput has a linear relationship to the offered load until saturation is reached, where the packet arrival rate is λ_{Sat}. Note λ_{Sat} has a significant effect on the goodput, as it relates to T_{SF}; the smaller the value of T_{SF} the higher the saturation goodput is, given that there is enough time for all nodes to transmit during the CFP.

Notice figures 7.1 and 7.3 illustrate the throughput when there are only pollable nodes in the superframe and the total number of nodes in the system is as indicated on the graph. The effect of this is there are no nodes operating in the contention period that could delay the start of the beacon frame transmission, which increases the time between consecutive beacon frames. This results in the agreement between the model and the simulated results.

Figures 7.2 and 7.4 illustrate the principle of nodes operating in the CFP and the CP. A total number of 10 nodes are in the system and the nodes operating in the CP have the same constant packet size and packet arrival rate as that of the nodes operating in the CFP. The average and relative percentage errors for this scenario increases because of the nodes operating in the CP, which increases the delay between beacon frames. The majority of the error increase is because of the changeover from the unsaturated to saturated region. When nodes operate in the CP, the start of beacons are delayed that results in the increase of T_{SF}, which in turn reduces λ_{Sat} . This reduction will cause the model to overestimate the throughput.

If fewer nodes are operating in the CFP, when there are nodes operating in the CP, for this simulation example more nodes will transmit in the CP. This increases the probability of transmissions delaying the start of the CFP and increases the error obtained. Note, also, as the number of pollable nodes increase, the expected service cycle time increases, which in turn decreases the available time for contention nodes to transmit in the CP, increasing the likelihood of nodes delaying the start of the CFP. This results in an increased error. These observations can be verified by referring to tables 7.1, 7.2, 7.3 and 7.4.

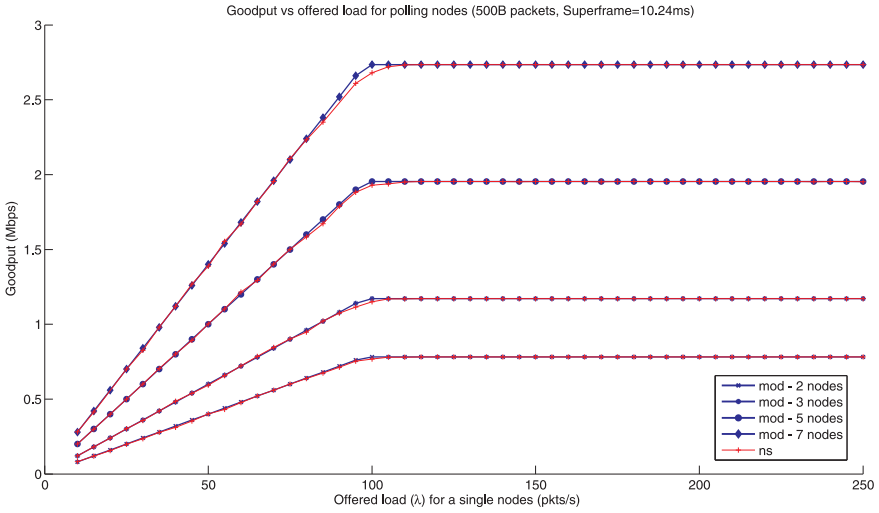


Figure 7.1: Total PCF goodput versus the packet arrival rate for a single node, with only a variable number of polling nodes (500B packet size, superframe=10.24ms)

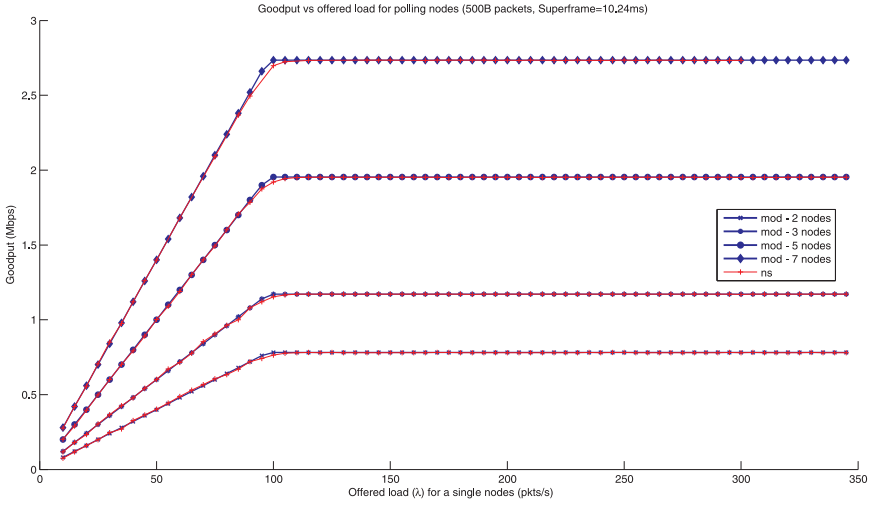


Figure 7.2: Total PCF goodput versus the packet arrival rate for a single node, with polling nodes and nodes operating in the CP (500B packet size, superframe=10.24ms)

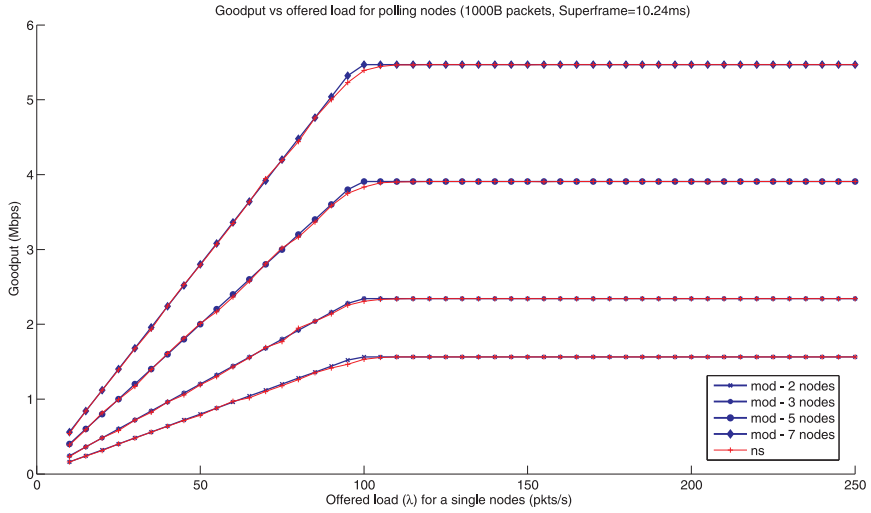


Figure 7.3: Total PCF goodput versus the packet arrival rate for a single node, with only a variable number of polling nodes (1000B packet size, superframe=10.24ms)

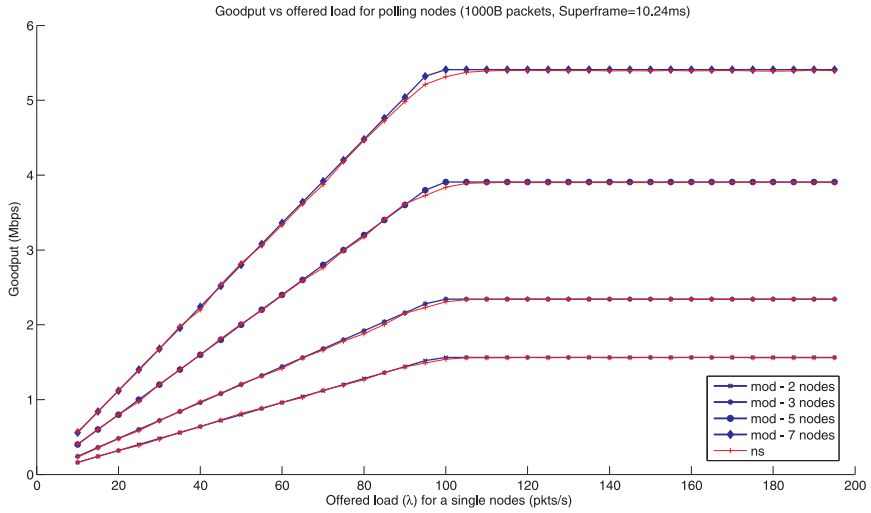


Figure 7.4: Total PCF goodput versus the packet arrival rate for a single node, with polling nodes and nodes operating in the CP (1000B packet size, superframe=10.24ms)

Table 7.1: Results summary for a variable number of pollabe nodes with superframe = 10.24ms and 500B packet size

Number of CFP nodes	Average error (kbps)	Percentage relative error	Average saturation throughput simulation (Mbps)	Average saturation through-put model (Mbps)
2	1.7112	0.26385	0.7813	0.7813
3	2.2405	0.23031	1.172	1.172
5	3.36	0.20723	1.953	1.953
7	6.1125	0.2747	2.735	2.735

Table 7.2: Results summary for a variable number of pollabe and CP nodes with superframe = 10.24ms and 500B packet size

Number of CFP nodes	Average error (kbps)	Percentage relative error	Average saturation throughput simulation (Mbps)	Average saturation through-put model (Mbps)
2	2.3638	0.34477	0.7806	0.7813
3	2.5901	0.25184	1.171	1.172
5	3.2757	0.19111	1.951	1.953
7	3.2427	0.14066	2.733	2.735

Table 7.3: Results summary for a variable number of pollabe nodes with superframe = 10.24ms and 1000B packet size

Number of CFP nodes	Average error (kbps)	Percentage relative error	Average saturation throughput simulation	Average saturation through-put model (Mbps)
2	5.3537	0.41274	1.563	1.563
3	6.2029	0.31881	2.344	2.344
5	9.9041	0.30542	3.906	3.906
7	8.8959	0.19595	5.469	5.469

Table 7.4: Results summary for a variable number of pollabe and CP nodes with superframe = 10.24ms and 1000B packet size

Number of CFP nodes	Average error (kbps)	Percentage relative error	Average saturation throughput simulation (Mbps)	Average saturation through-put model (Mbps)
2	4.854	0.39778	1.561	1.563
3	8.7988	0.4807	2.343	2.344
5	11.27	0.36944	3.9	3.906
7	23.642	0.55756	5.399	5.469

7.4 PCF packet delay for unsaturated conditions

In this section a method described by [Sikbar-05] which will be referred to as Sikbar's method will be outlined to determine the queueing delay for PCF nodes operating under unsaturated conditions.

7.4.1 Assumption

No stations are using power save mode. Power save mode is a mode in which if a station has no packets to send, it goes into sleep mode. The PC buffers frames and notifies nodes in power save mode when they should awake to receive these packets by sending this information during the beacon frame transmission, using traffic indication map (TIM).

7.4.2 Arrivals at an empty queue

Referring to figure 7.5, which shows the time evolution of transmissions on the channel, the tagged poll is indicated for a specific tagged node which is t_1 seconds after the beacon is sent and the CFP started. If a packet arrives at an empty queue for a pollable node before the tagged poll, the packet will be transmitted during the current superframe. However, if the packet arrives after the tagged node has been polled, the node will only be serviced during the next CFP. An example of this is the arrival t_2 seconds after the CFP is started. Note figure 7.5 does actually show that no packet arrival occurred before the poll was received for the tagged node. As no data packet was transmitted, a PIFS time period passed before the next poll was sent.

The two scenarios to consider are when a packet arrives at an empty queue before it is polled in the current superframe and when it arrives after being polled. Given that the packet arrival occurs in the current superframe, the assumption is

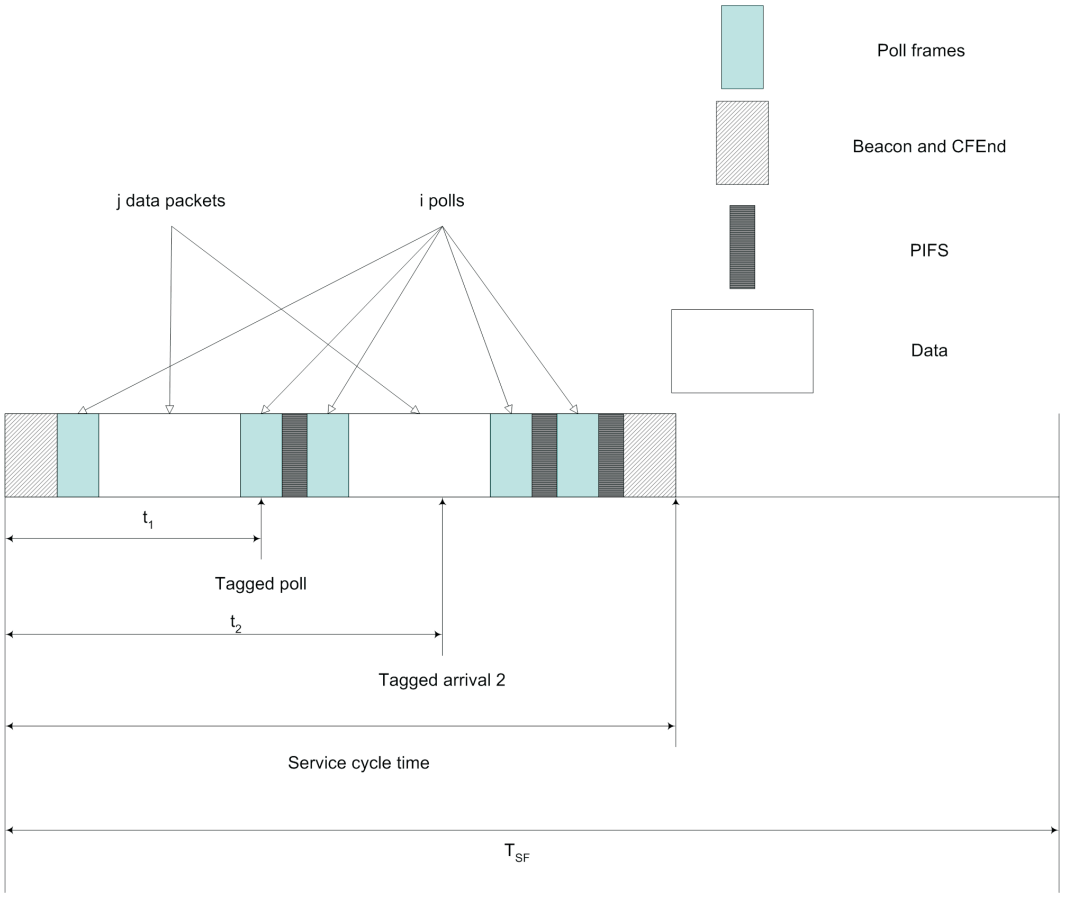


Figure 7.5: Packet delay when arriving packet finds queue empty (Case C2)

that packet arrival is a uniformly distributed random variable in the range of $[0, T_{SF}]$ relative to the start of the superframe.

7.4.3 Packet arriving at an empty queue

Suppose our tagged node is the ith node on the polling list and j nodes transmit data packets before this node is polled. For this scenario a period of $B + (i-1)V + jL + (i-j-1)T_{PIFS}$ seconds elapse before the ith node is polled and $B + iV + jL + (i-j)T_{PIFS}$ seconds elapse before the node has to respond to the poll. Thus, if the tagged arrival occurs during this time interval, it will be serviced during this superframe. Otherwise it will have to wait for the next superframe.

Since the packet arrival instant t is relative to the start of the superframe and modelled as a uniformly distributed random variable $(U[0, T_{SF}])$, the probability of a tagged arrival occurring for node i in the first $B + iV + jL + (i-j)T_{PIFS}$ seconds is

$$P[t \leq B + iV + jL + (i-j)T_{PIFS}] = \frac{B + iV + jL + (i-j)T_{PIFS}}{T_{SF}} \tag{7.4.1}$$

Let the packet arrival occurring in the current superframe at an empty queue and serviced in the current frame be referred to as case $C1$. The time a packet waits before it is serviced is $X_{i,j,C1} = B + iV + jL + (i-j)T_{PIFS} - t$, after which it receives service for another L seconds before departing the system. The probability distribution function of t, given that the packet did arrive in the first $B + iV + jL + (i-j)T_{PIFS}$ seconds of the current superframe, is

$$\begin{aligned} P[t \leq \tau | t \leq B + iV + jL + (i-j)T_{PIFS}] &= \frac{P[t \leq \tau, t \leq B + iV + jL + (i-j)T_{PIFS}]}{P[t \leq B + iV + jL + (i-j)T_{PIFS}]} \\ &= \frac{\tau}{B + iV + jL + (i-j)T_{PIFS}} \end{aligned} \tag{7.4.2}$$

which is uniform distribution in the range $[0, B + iV + jL + (i-j)T_{PIFS}]$. Note that if a random variable Y is uniformly distributed in the range 0 to a, the random variable $a - Y$ is also uniformly distributed in the range 0 to a. Using this observation, the conditional probability t is uniformly distributed in the range 0 to $B + iV + jL + (i-j)T_{PIFS}$, which makes the conditional probability $X_{i,j,C1} = B + iV + jL + (i-j)T_{PIFS} - t$ also uniformly distributed for the same range. The expected value for $X_{i,j,C1}$ is thus

$$E[X_{i,j,C1}] = E[U[0, B + iV + jL + (i-j)T_{PIFS}]] = \frac{B + iV + jL + (i-j)T_{PIFS}}{2} \tag{7.4.3}$$

When the packet arrives after the first $B + iV + jL + (i-j)T_{PIFS}$ seconds of the current superframe, it will be referred to as case $C2$. This implies $t > B + iV + jL + (i-j)T_{PIFS}$, the packet has to wait for the remainder of the current superframe $(T_{SF} - t)$ and until node i is polled in the next superframe. The PDF, given the packet arrival occurred after the first $B + iV + jL + (i-j)T_{PIFS}$ seconds, is then also uniformly distributed in the range $[B + iV + jL + (i-j)T_{PIFS}, T_{SF}]$. Thus the remaining part of the superframe, $T_{SF} - t$, is also uniformly distributed as $[0, T_{SF} - B - iV - jL - (i-j)T_{PIFS}]$.

In the following superframe, if there are k nodes with data to transmit before the ith node is polled, the tagged node has to wait $B + iV + jL + (i-j)T_{PIFS}$ second before it is serviced. Since the probability exists that there are k nodes with data to transmit among the i nodes, the probability mass function (pmf) of the waiting

time is modelled as binomial distribution as

$$P[X_{SC_N} = x] = \begin{cases} \binom{i}{k} \rho^k (1-\rho)^{i-k} & x = B + iV + kL + (i-k)T_{PIFS} \\ 0 & otherwise \end{cases} \tag{7.4.4}$$

with $0 \leq k \leq i$ and the expected value of the waiting time

$$E[X_{SC}] = B + iV + ipL + i(1-p)T_{PIFS} \tag{7.4.5}$$

Thus the time a packet has to wait for service for case $C2(X_{i,j,C2})$ is $X_{i,j,C2} = T_{SF} - t + X_{SC}$. The expected value of $X_{i,j,C2}$ is thus

$$\begin{aligned} E[X_{i,j,C2}] &= E[U[0, T_{SF} - B - iV - jL - (i-j)T_{PIFS}]] + E[X_{SC_N}] \\ &= \frac{T_{SF} - B - iV - jL - (i-j)T_{PIFS}}{2} + B + iV + i\rho L + (1-\rho)i \cdot T_{PIFS} \end{aligned} \tag{7.4.6}$$

The expected waiting time for a packet arriving at an empty queue, given that j out of i nodes have packets to transmit and before the ith node can transmit data in the current superframe, cases $C1$ and $C2$ have to be combined. This expected waiting time $D_{i,j,EQ}$ is given by

$$D_{i,j,EQ} = E[X_{i,j}] + L \tag{7.4.7}$$

With

$$E[X_{i,j}] = E[X_{i,j,C1}]P[C1] + E[X_{i,j,C2}]P[C2] \tag{7.4.8}$$

where $X_{i,j,C1}$ and $X_{i,j,C2}$ is given by equation 7.4.3 and equation 7.4.6. $P[C1]$ and $P[C2]$ are the probabilities of a packet arriving in the first $B + iV + jL + (i-j)T_{PIFS}$ seconds of the superframe or not, respectively. These probabilities are

$$\begin{aligned} P[C1] &= \frac{B + iV + jL + (i-j)T_{PIFS}}{T_{SF}} \\ P[C2] &= 1 - \frac{B + iV + jL + (i-j)T_{PIFS}}{T_{SF}} \end{aligned} \tag{7.4.9}$$

Putting these values into equation 7.4.8, $E[X_{i,j}]$ is simplified to

$$E[X_{i,j}] = \frac{T}{2} + \frac{(B+iV+jL+(i-j)T_{PIFS})^2}{T_{SF}} - (B+iV+jL+(i-j)T_{PIFS}) \\ +E[X_{SC_N}]\frac{T_{SF}-B-iV-jL-(i-j)T_{PIFS}}{T_{SF}} \tag{7.4.10}$$

The expected delay of the ith node $D_{i,j,EQ}$ is obtained by substituting equation 7.4.7 in $(E[X_{i,j}])$. To determine the delay for the system $D_{i,j,EQ}$ has to be unconditioned from j to obtain $D_{i,EQ}$. Note that j is the number of nodes before the ith with data packets to transmit and is modelled as a binomial distribution given by equation 7.4.4. Thus $D_{i,EQ}$ is given by

$$\begin{aligned} D_{i,EQ} &= \sum_{j=0}^{i}(E[X_{i,j}]+L)\binom{i}{j}\rho^j(1-\rho)^{i-j} \\ &= \frac{T_{SF}}{2} + \frac{i\rho(L^2+T_{PIFS}^2)(1-\rho)}{T_{SF}} + L \end{aligned} \tag{7.4.11}$$

Comparing equation 7.4.11 with the one used in [Sikbar-05], it can be noticed the equation used was slightly changed to compensate for the fact that the BS only transmits control frames, and the inclusion of T_{PIFS} for nodes with no data frames to transmit.

7.4.4 Packets arriving at a non-empty queue

We now consider the case where an arbitrary packet arrival occurs at the ith node when the queue is non-empty. The number of packets in the queue is denoted as N_{NQ}. In this case, the packet has to wait for all the other packets in the queue to be serviced before it can start service. Calculating the expected waiting time again has two cases, (1) where the packet arrival occurs before the ith node is serviced in the current superframe (case $C1$), and 2) when the node has already been serviced (case $C2$). Of the i nodes, there are j nodes that have data packets to transmit before the ith node is polled in the current superframe. Then the probability of events $C1$ and $C2$ occurring is

$$P[C1] = \frac{B+iV+(j+1)L+(i-j-1)T_{PIFS}}{T_{SF}}$$
$$P[C2] = 1 - \frac{B+iV+(j+1)L+(i-j-1)T_{PIFS}}{T_{SF}} \tag{7.4.12}$$

where the $j+1$ term indicates that besides the j nodes that are transmitting, node i also has a packet to transmit because of its non-empty queue.

In case $C1$ where the ith node has not been served when the tagged arrival occurs, one of the N_{NQ} packets currently waiting in the queue will be serviced. If we denote the packet arrival instant by t, then the packet has to wait another $T_{SF}-t$ seconds for the next superframe, where it again has to wait for another $N_{NQ}-1$ packets to depart. Therefore, the packet has to wait another $(N_{NQ}-1)T_{SF}$ seconds before the superframe starts in which it will be serviced. The final waiting time of the superframe is defined as X_{SC_F}. Thus the total waiting time for the packet for case $C1$ is given by

$$X_{i,j,C1} = T_s - t + (N_{NQ}-1)T_s + X_{SC_F} \tag{7.4.13}$$

The probability that the arrival occurs in the first $B+iV+(j+1)L+1(i-j-1)T_{PIFS}$ seconds is a uniform distribution with the range $[0, B+iV+(j+1)L+(i-j-1)T_{PIFS}]$, with $T_{SF}-t$ is in the range $[T_{SF}-B-iV-(j+1)L-(i-j-1)T_{PIFS}, T_{SF}]$. To evaluate the distribution of X_{SC_F}, we note that there are k nodes that have data to transmit before the ith node, and the packet has to wait $B+iV+kL+(i-k)T_{PIFS}$ seconds before service begins. Since k follows a binomial distribution, the probability mass function (pmf) is given by

$$P[X_{SC_F}=x] = \begin{cases} \binom{i}{k}\rho^k(1-\rho)^{i-k} & x = B+iV+kL+(i-k)T_{PIFS} \\ 0 & otherwise \end{cases} \tag{7.4.14}$$

where $0 \le k \le i$ and the expected value for X_{SC_F} is

$$E[X_{SC_F}] = B+iV+i\rho L+(1-\rho)i\cdot T_{PIFS} \tag{7.4.15}$$

The expected value of $X_{i,j,C1}$ is thus

$$\begin{aligned} E[X_{i,j,C1}] &= E[T_S - t] + E[(N_{NQ} - 1)T_S] + E[X_{FR}] \\ &= \frac{B + iV + (j+1)L + (i-j-1)T_{PIFS}}{2} + (E[N_{NQ}] - 1)T_S \\ &+ B + iV + i\rho L + (1-\rho)iL \cdot T_{PIFS} \end{aligned} \tag{7.4.16}$$

For case $C2$ the tagged packet arrival occurs when the ith node has already been serviced in the current superframe and the buffer still has N_{NQ} packets to be serviced. Thus, a further $N_{NQ}T_{SF}$ seconds is required for the start of the superframe where service takes place. Again, the expected service time $E[X_{SC_F}]$ is given by equation 7.4.14 and equation 7.4.15. The total time before service begins for case $C2$ is given by

$$X_{i,j,C2} = T_s - t + N_{NQ}T_s + X_{SC_F} \tag{7.4.17}$$

Now, the PDF that the packet arrival occurred after the first $B + iV + (j+1)L + (i - j - 1)T_{PIFS}$ seconds of the current superframe is

$$P[t \leq \tau | t > B + iV + (j+1)L + (i-j-1)T_{PIFS}] = \frac{\tau}{B + iV + (j+1)L + (i-j-1)T_{PIFS}} \tag{7.4.18}$$

with the uniform distributions range being $[T_{SF} - B - iV - (j+1)L - (i-j-1)T_{PIFS}, T_{SF}]$. Therefore, $T_{SF} - t$ is also uniformly distributed as $U[0, T_{SF} - B - iV - (j+1)L - (i-j-1)T_{PIFS}]$. The expected value of $X_{i,j,C2}$ is

$$\begin{aligned} X_{i,j,C2} &= E[T_s - t] + N_{NQ}T_s + X_{SC_F} \\ &= \frac{T_S - B - iV - (j+1)L - (i-j-1)T_{PIFS}}{2} + E[N_{NQ}]T_s \\ &+ B + iV + (j+1)L + (i-j-1)T_{PIFS} \end{aligned} \tag{7.4.19}$$

Combining the probabilities for cases $C1$ and $C2$, the expected waiting time at the ith node, $D_{i,j,NEQ}$ is given by

$$\begin{aligned} D_{i,j,NEQ} &= E[X_{i,j}] + L \\ &E[X_{i,j,C1}]P[C1] + E[X_{i,j,C2}]P[C2] + L \\ &= \frac{T_S}{2} + E[N_{NQ}]T_S + E[X_{SC_F}] - B - iV - jL - (i-j)T_{PIFS} \end{aligned} \tag{7.4.20}$$

Unconditioning the above equation of j and recalling j is modelled according to a

binomial distribution, the expected delay at the *ith* node, $D_{i,NEQ}$ is given by

$$\begin{aligned} D_{i,NEQ} &= \sum_{j=0}^{i} D_{i,j,NEQ} \binom{i}{j} \rho^j (1-\rho)^{i-j} \\ &= \frac{T_{SF}}{2} + E[N_{NQ}] T_S \end{aligned} \tag{7.4.21}$$

7.4.5 Overall delay

The delays of the previous two sections can now be combined to obtain the overall delay for a packet for a node operating in PCF mode. The empty queue probability is

$$P[EQ] = 1 - \rho = 1 - \lambda T_{SF} \tag{7.4.22}$$

where ρ is the utilization. The probability that a queue is not empty is

$$P[NEQ] = 1 - P[EQ] = \rho = \lambda \cdot T_{SF} \tag{7.4.23}$$

Now, the overall delay experienced by a node operating in active mode is

$$\begin{aligned} D_{i,AM} &= D_{i,EQ} P[EQ] + D_{i,NEQ} P[NEQ] \\ &= \frac{T_S}{2} + \rho E[N_{NQ}] T_S + \left[\frac{i\rho(L^2 + T_{PIFS}^2)(1-\rho)}{T_S} + L \right] (1-\rho) \end{aligned} \tag{7.4.24}$$

where $P[EQ]$, $P[NEQ]$, $D_{i,NEQ}$ and $D_{i,EQ}$ is given by equations 7.4.22, 7.4.23, 7.4.11 and 7.4.21. Note, however, that the expression $E[N_{NQ}]$ is the expected number of packets seen by an arriving packet when the queue is not empty. The expected number of packets in the queue seen by an arbitrary arrival is $E[N] = \sum_{i=0}^{\infty} iP[N=i]$, and is related to $E[N_{NQ}]$ by

$$E[N_{NQ}] = \sum_{i=0}^{\infty} \frac{iP[N=i,NEQ]}{P[NEQ]} = \sum_{i=0}^{\infty} \frac{iP[N=i]}{\rho} = \frac{E[N]}{\rho} \tag{7.4.25}$$

$P[N=I,NEQ]$ represents the joint probability of i packets in a queue, given that the queue is non-empty. Also, from Little's law $E[N] = \lambda \cdot D_{i,AM}$ resulting in $E[N_{NQ}] = \lambda \cdot D_{i,AM}/\rho$, substituting this into equation 7.4.24 gives the final expression

$$D_{i,AM} = \frac{1}{1-\lambda T_S}\left[\frac{T_S}{2} + \left(\frac{i\rho(L^2+T_{PIFS}^2)(1-\rho)}{T_S} + L\right)(1-\rho)\right] \tag{7.4.26}$$

As this is a system where nodes have finite buffers, another adjustment made to the authors original model was to include the blocking probability. The blocking probability P_B is defined as (see Hok-97)

$$P_b = \frac{(1-\rho)p^K}{1-\rho^{K+1}} \tag{7.4.27}$$

where ρ is the utilization and K is the maximum number of packets the buffer accommodates. Since customers are blocked when there are K packets in the buffer, the effective packet arrival rate of the packets admitted into the system is

$$\lambda^{'} = \lambda(1-P_b) \tag{7.4.28}$$

The utilization is $\rho = \lambda T$ and the effective utilization becomes

$$\rho^{'} = \rho(1-P_b) \tag{7.4.29}$$

By substituting the above equation into equation 7.4.26 we obtain $D'_{i,AM}$ as

$$D_{i,AM}^{'} = \frac{1}{1-\rho^{'}}\left[\frac{T_S}{2} + \left(\frac{i\rho^{'}(L^2+T_{PIFS}^2)(1-\rho^{'})}{T_S} + L\right)(1-\rho^{'})\right] \tag{7.4.30}$$

Comparing equation 7.4.30 with the one used in [Sikbar-05], it can be noticed the equation used was slightly changed to compensate for the fact that the BS only transmits control frames and no data frames, and the inclusion of T_{PIFS} for nodes with no data frames to transmit. Also notice, the inclusion of the blocking probability.

7.4.6 Average buffer occupancy

The average buffer occupancy is obtained by using equation 7.4.30 and by using Little's law, the utilization and the probability that the queue is not empty. As stated

in the previous section $E[N_{NQ}] = \lambda \cdot D_{i,AM}/\rho$ is the expected number of packets seen by an arriving packet when the queue is not empty. Also, from equation 7.4.25 where we know $E[N] = E[N_{NQ}] \cdot \rho$. Note as we are using a system with finite buffer space, the effective packet arrival and utilization rates will be used with

$$E^{'}[N_{NQ}] = \lambda^{'} \cdot D_{i,AM} \tag{7.4.31}$$

And

$$E^{'}[N] = E^{'}[N_{NQ}] \cdot \rho^{'} \tag{7.4.32}$$

7.5 PCF packet delay for the saturated condition

The saturated condition is defined here as every PCF node having a packet ready to be transmitted as soon as the previous packet has been transmitted and the utilization being greater than or equal to one ($\rho \geq 1$). Queueing theory would indicate this to be a $M/D/1/k$ system (see: [Hoc-97]), where the arrival process is Markovian, the service process is deterministic, only one server is available and k indicates the finite buffer capacity.

As the PCF mode enters saturation, the service process becomes roughly deterministic because all nodes always have a packet to transmit and a constant superframe period ensures that the service time remains constant. We assume the service process to be deterministic, although it is not always the case. Sometimes the interval between beacons might be longer because a DCF node started transmitting slightly before the beacon. Hence, this might delay the proper start of the beacon. However, it will be proved this has little effect on the results.

A special case of a $M/D/1/k$ queue, in fact, is the $M/M/1/k$ queue. The service time for the M/M/1 queue uses the average service time which, if considered as constant, is actually deterministic.

7.5.1 Queueing theory delay analysis

Following the analysis for $M/M/1/k$ queues, the blocking probability P_B is defined as in equation 7.4.27. Using the effective packet arrival rate of the packets admitted into the system the average time a packet spends in the system is

$$T = \frac{N}{\lambda'} = \frac{1}{\mu - \lambda'} - \frac{K\rho^{K+1}}{\lambda' - \mu\rho^{K+1}} \tag{7.5.1}$$

where $\mu = 1/T_{SF}$ is the service rate and λ' the effective packet arrival rate. The average number of customers (N) in the system is given by

$$N = \frac{\rho}{1-\rho} - \frac{\rho}{1-\rho}(K+1)P_b \tag{7.5.2}$$

7.6 Combining the buffer occupancy/ delay for the saturated and non-saturated condition

To obtain the delay and buffer occupancy, the approach for non-saturated conditions will be used when the utilization is less than one, that is $\rho < 1$ (see section 7.4). Whereas, for $\rho \geq 1$ the approach for saturated conditions is used (see section 7.5).

7.7 Results

7.7.1 End-to-end delay results

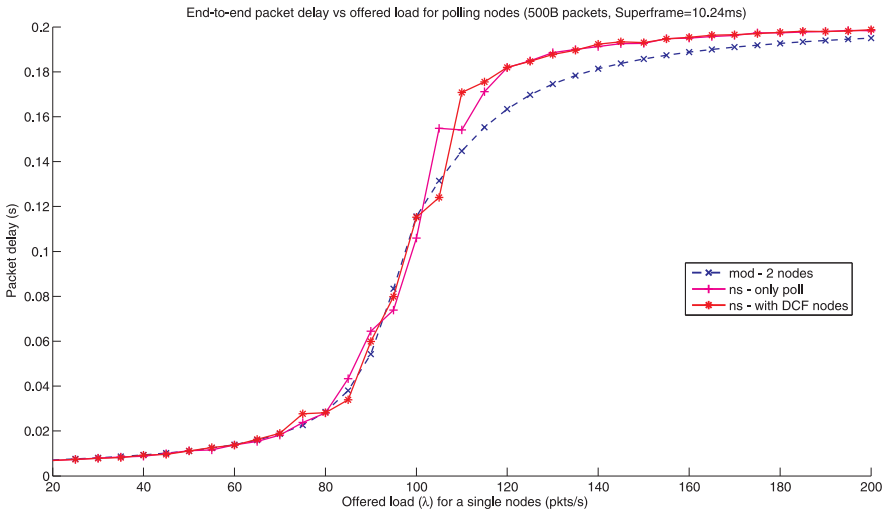


Figure 7.6: Average end-to-end delay versus the offered load for 2 polling nodes with 500B packets and T_{SF}=10.24ms.

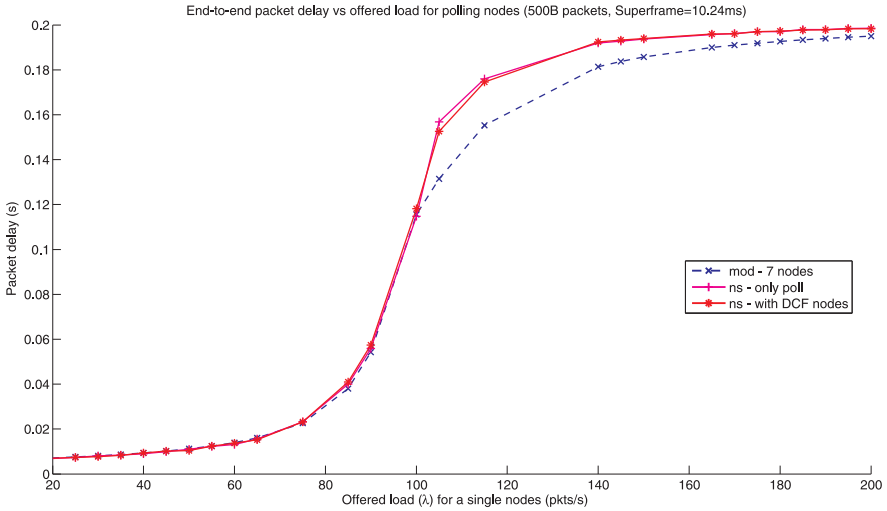


Figure 7.7: Average end-to-end delay versus the offered load for 7 polling nodes with 500B packets and T_{SF}=10.24ms.

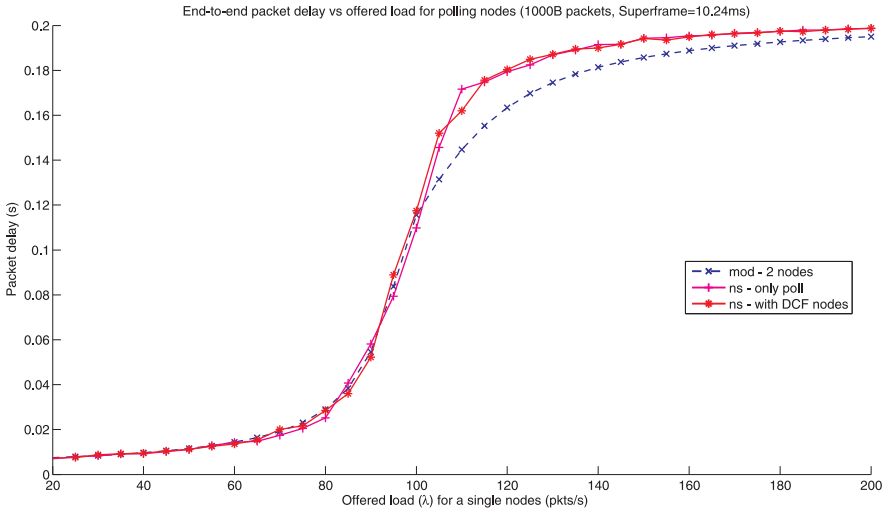


Figure 7.8: Average end-to-end delay versus the offered load for 2 polling nodes with 1000B packets and T_{SF}=10.24ms.

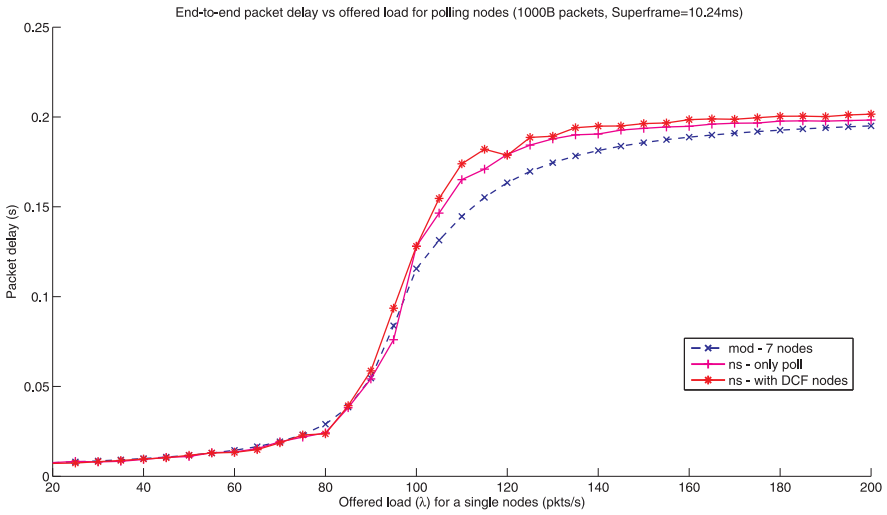


Figure 7.9: Average end-to-end delay versus the offered load for 7 polling nodes with 1000B packets and T_{SF}=10.24ms.

Table 7.5: Average end-to-end delay results for 500B packets and Tsf=10.24ms (only pollable nodes)

Number of polling nodes	Average error (ms)	Percentage error (%)	Average saturation delay simulation (s)	Average saturation delay model (ms)
2	5.1173	4.1551	0.2002	0.1982
3	4.7275	3.8382	0.2003	0.1982
5	3.9899	3.9899	0.20022	0.1982
7	3.8001	4.3023	0.20022	0.1982

Table 7.6: Average end-to-end delay results for 500B packets and Tsf=10.24ms (with CP nodes)

Number of polling nodes	Average error (ms)	Percentage error (%)	Average saturation delay simulation (s)	Average saturation delay model (ms)
2	4.9123	3.9886	0.2003	0.1982
3	5.0138	4.0707	0.2003	0.1982
5	3.5667	3.5667	0.20024	0.1982
7	3.7713	4.3023	0.20025	0.1982

Table 7.7: Average end-to-end delay results for 1000B packets and Tsf=10.24ms (Only pollable nodes)

Number of polling nodes	Average error (ms)	Percentage error (%)	Average saturation delay simulation (s)	Average saturation delay model (s)
2	4.9964	4.0519	0.2001	0.198
3	5.0323	4.0799	0.2004	0.1982
5	5.0702	4.1106	0.20039	0.1982
7	4.9263	3.9939	0.20042	0.1982

Table 7.8: Average end-to-end delay results for 1000B packets and Tsf=10.24ms (With CP nodes)

Number of polling nodes	Average error (ms)	Percentage error (%)	Average saturation delay simulation (s)	Average saturation delay model (s)
2	4.7545	2.603	0.2001	0.198
3	4.533	3.675	0.20043	0.1982
5	4.9176	3.9869	0.20039	0.1982
7	7.1436	5.7915	0.2027	0.1982

7.7.2 Discussion of end-to-end delay results

In the previous section the results show that the end-to-end packet delay for nodes operating in PCF mode is well approximated with the mathematical model. From Tables 7.5 to 7.8, the average saturation throughput for the mathematical model all saturate to the same value which is $0.1982ms$. The $M/M/1/k$ queueing model used for the saturated region computes the packet delay with equation 7.5.1, which is independent of the number of nodes communicating in the CFP. The simulations, do however, indicate there is a slight increase in saturation delay when the number of nodes transmitting in the CFP increases. The model used for the unsaturated region also proves to be a good prediction of the average end-to-end delay. Note, also, that the delay in the unsaturated condition shows insignificant change in the number of pollable nodes. In the changeover region which we define as the region between the saturated and unsaturated region, we find the model underestimates the delay obtained by *ns*.

Figures 7.6 to 7.9 indicates the average end-to-end delay for a packet that operates in a RRP PCF network, and on each figure the results obtained from the mathematical model and from simulation are indicated. Only pollable nodes are indicated by "*ns* - only poll", which implies no nodes are operating in the CP and "with DCF nodes" implies there are nodes operating in the CFP and CP.

Note, the "with DCF nodes" proves to be slightly more underestimated in the changeover region and should also be for the saturation region. The reason for this is the model does not compensate for the chance that nodes operating in the CP might transmit a packet that will delay the start of the CFP. Thus in effect this increases the average duration of the superframe. Also, this is the reason why the model underestimates the delay in the saturated region.

7.7.3 Average buffer occupancy results

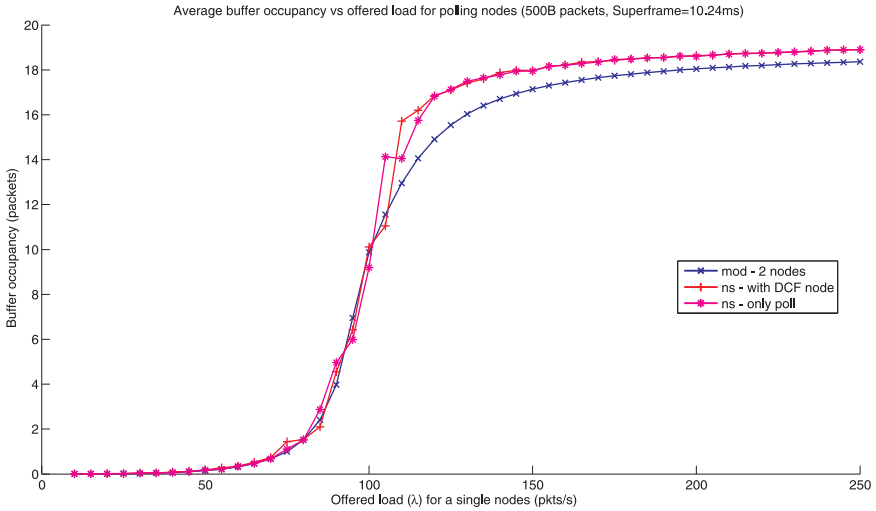


Figure 7.10: Average buffer occupancy versus the offered load for 2 polling nodes with 500B packets and T_{SF}=10.24ms.

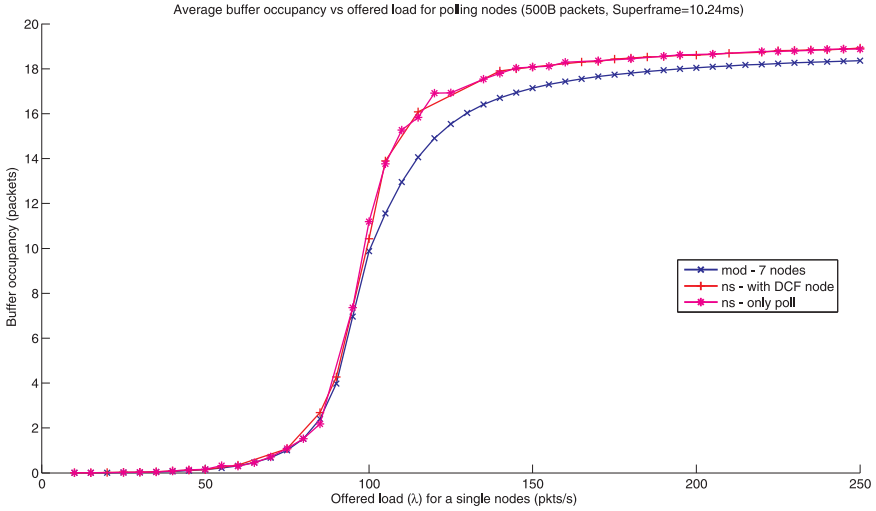


Figure 7.11: Average buffer occupancy versus the offered load for 7 polling nodes with 500B packets and T_{SF}=10.24ms.

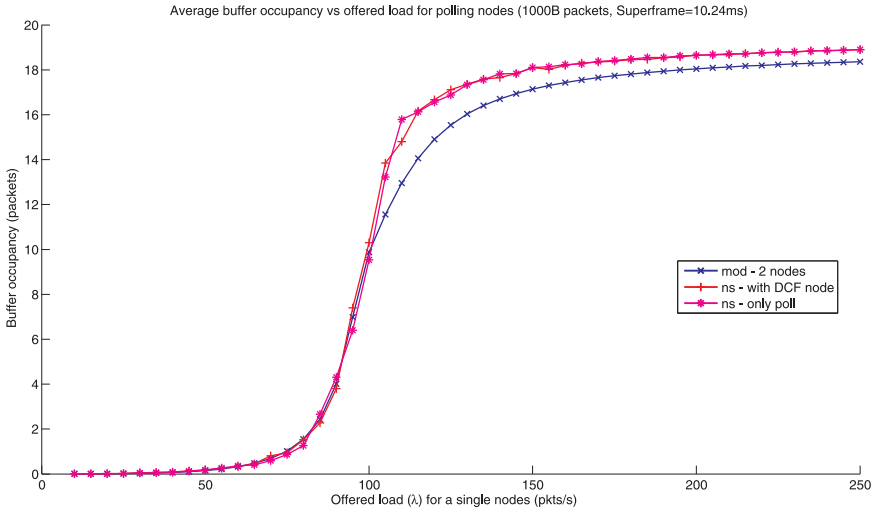


Figure 7.12: Average buffer occupancy versus the offered load for 2 polling nodes with 1000B packets and T_{SF}=10.24ms.

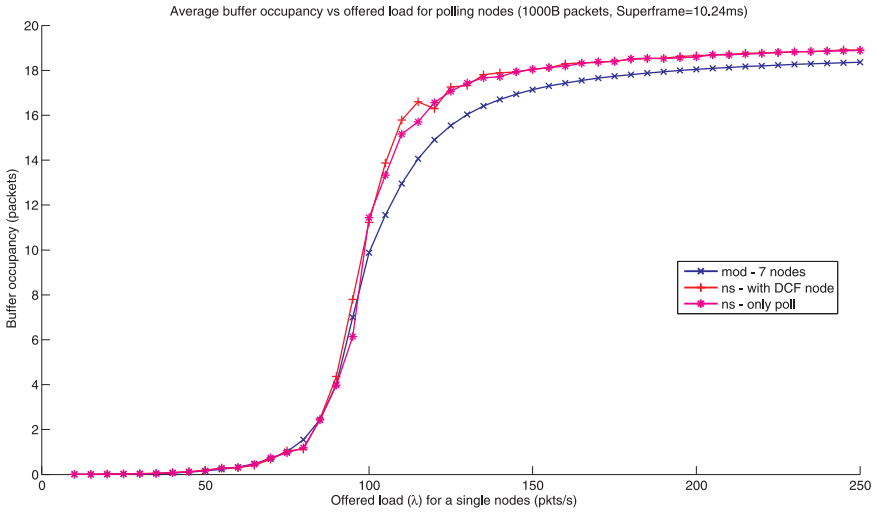


Figure 7.13: Average buffer occupancy versus the offered load for 7 polling nodes with 1000B packets and T_{SF}=10.24ms.

Table 7.9: Average buffer occupancy results for 500B packets and T_{SF}=10.24ms (only pollable nodes)

Number of polling nodes	Average error (packets)	Percentage error (%)
2	0.61651	5.5968
3	0.67858	6.1602
5	0.60968	5.5347
7	0.62142	6.0297

Table 7.10: Average buffer occupancy results for 500B packets and T_{SF}=10.24ms (with CP nodes)

Number of polling nodes	Average error (packets)	Percentage error (%)
2	0.58894	5.4556
3	0.59831	5.5424
5	0.54522	5.0505
7	0.51475	4.7472

Table 7.11: Average buffer occupancy results for 1000B packets and T_{SF}=10.24ms (only pollable nodes)

Number of nodes	Average error (packets)	Percentage error (%)
2	0.60494	5.4897
3	0.60772	5.5149
5	0.62928	5.7104
7	0.60755	5.5131

Table 7.12: Average buffer occupancy results for 1000B packets and T_{SF}=10.24ms (with CP nodes)

Number of nodes	Average error (packets)	Percentage error (%)
2	0.5714	5.2912
3	0.54068	5.0067
5	0.58545	5.4211
7	0.65061	6.0243

7.7.4 Discussion of results

As the packet delay and the buffer occupancy are related, the buffer occupancy and packet delay results in Sections 7.7.1 and 7.7.2 are in agreement. Evidently the number of pollable nodes does not have a significant effect on the results and equation 7.5.2 states that the saturation buffer occupancy is independent of the number of nodes. An average relative error of roughly six percent is achieved, which proves this model to be relatively accurate.

Little's law predicts the packet delay and buffer occupancy curves to conform to each other. Since the packet delay obtained by mathematical modelling is underestimated in the saturation region, so will the buffer occupancy.

7.8 Summary

In this section a mathematical approach to model the service cycle time, throughput, end-to-end delay and buffer occupancy of nodes operating in a RRP scheme in conjunction with the PCF mode was discussed. It was found that the simple method devised to determine the throughput proved to be very accurate, with an average relative error of roughly one percent. The average end-to-end packet delay and buffer occupancy was also found to be modelled accurately with an average relative error of six percent.

Chapter 8

Combined DCF mathematical modelling

In this chapter a combined mathematical model will be discussed, where the principles introduced in chapters 6 and 7 are combined to model an IEEE 802.11 network operating in PCF mode. The approach of determining the throughput and delay of only DCF nodes operating in the contention period with a superframe structure will be discussed. The analysis of PCF nodes operating in a superframe structure was discussed in chapter 7.

8.1 Scaling throughput

As discussed in section 5.4, the DCF nodes pause their backoff timers when a CFP starts. When the CP restarts (after the reception of the CFEnd) the backoff timers resume counting down from the same value they had when they were paused. Section 5.4.5.3 illustrated, by method of simulation the combined throughput for a superframe is the superposition of the throughput of the respective nodes transmitting in the CFP and CP.

The approach devised for this project is to use the expected service cycle to determine the values of the fractions for which the superframe will be occupied by the CFP and CP regions. By using the fraction values the throughput, delay and buffer occupancy are calculated.

The expected service cycle time as discussed in section 7.2 is repeated here and is defined as

$$E[X_{SC}] = B + iV + i\rho L + i(1-\rho)T_{PIFS} \tag{8.1.1}$$

where B compensates for the time it takes to transmit a beacon and CFEnd frame, V the time for a poll transmision, ρ the utilization, i the total number of pollable nodes

and T_{PIFS} the time for a PIFS. Note that the maximum value for ρ is one. When this value is reached the maximum service cycle time is obtained, where every node polled has a packet to transmit. If $\rho > 1$, the service cycle time will be set to $\rho = 1$ as the service cycle time can't increase above one.

8.1.1 Scaling DCF throughput

As discussed in Chapter 2 and shown in Chapters 6.5.2, with nodes operating in DCF mode, throughput for unsaturated conditions increases roughly linearly for an increasing traffic load until throughput saturation is reached. The expected percentage of the superframe occupied by the contention period is defined as

$$E[\%_{CP}] = (T_{SF} - E[X_{SC}])/T_{SF} \tag{8.1.2}$$

where T_{SF} is the duration of the superframe and $E[X_{SC}]$ is the expected service cycle time.

The throughput of contention nodes operating in the contention period matched the throughput of DCF nodes operating normally without any pollable nodes in the unsaturated region closely. However, the saturation throughput was less than with nodes operating only in DCF mode. This fact led to a new method being devised for this project for the new expected saturation throughput ($E[Tput_{NSAT}]$):

$$E[Tput_{NSAT}] = E[\%_{CP}] \cdot E[Tput_{SAT}] \tag{8.1.3}$$

The procedure is thus to use the same expected saturation throughput $E[Tput]$ value as obtained from Garetto's model but, as soon as the throughput exceeds $E[Tput_{NSAT}]$, it is limited to the new expected saturation value $E[Tput_{NSAT}]$.

8.1.2 Results

Table 8.1: Results for Figure 8.1

Number of CFP nodes	Average error (kbps)	Percentage error	Average saturation throughput/simulation (Mbps)	Average saturation throughput model (Mbps)
2	42.166	1.996	4.473	4.305
5	34.072	1.2156	3.509	3.442
7	21.9	1.0953	2.814	2.733

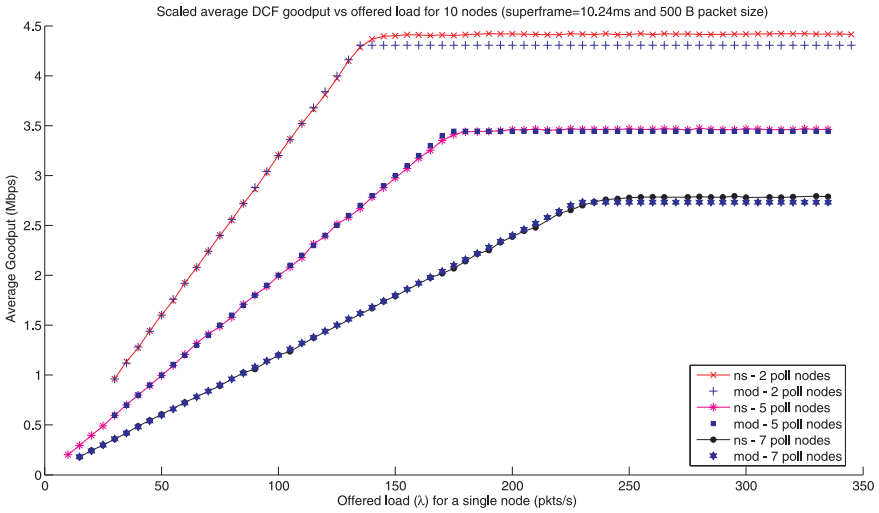


Figure 8.1: Comparison of scaled DCF goodput versus the average packet arrival where the number of polling nodes is varied (10 nodes with 500B packet sizes and $T_{SF} = 10.24ms$)

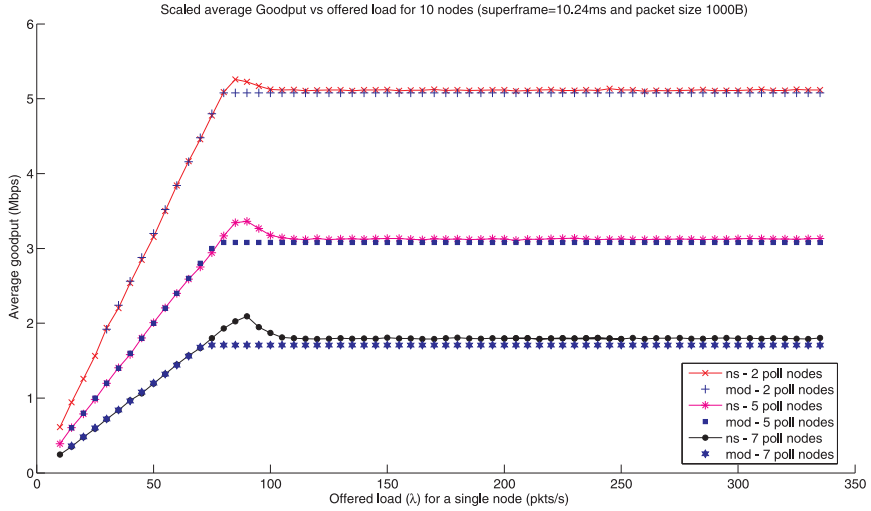


Figure 8.2: Comparison of scaled DCF goodput versus the average packet arrival where the number of polling nodes is varied (10 nodes with 1000B packet sizes and $T_{SF} = 10.24ms$)

Table 8.2: Results for Figure 8.2

Number of CFP nodes	Average error (kbps)	Percentage error	Average saturation throughput simulation (Mbps)	Average saturation throughput model (Mbps)
2	196.74	3.9122	5.116	5.078
5	63.203	1.798	3.199	3.078
7	54.116	2.274	1.856	1.709

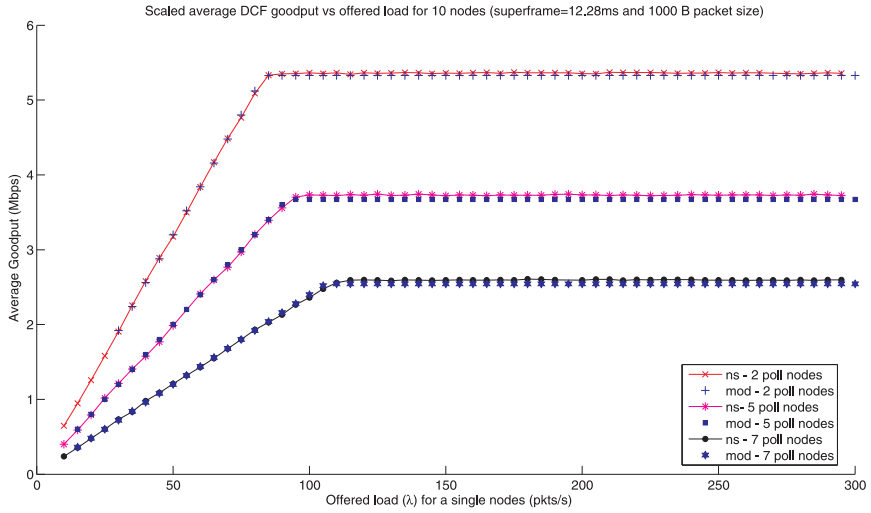


Figure 8.3: Comparison of scaled DCF goodput versus the average packet arrival where the number of polling nodes is varied (10 nodes with 1000B packet sizes and $T_{SF} = 12.28ms$)

Table 8.3: Results for Figure 8.3

Number of CFP nodes	Average error (kbps)	Percentage error	Average saturation throughput simulation (Mbps)	Average saturation throughput model (Mbps)
2	28.781	0.58113	5.36	5.328
5	47.885	1.4743	3.733	3.67
7	38.004	1.7523	2.594	2.543

8.1.3 Discussion

The results prove this to be an accurate estimation. From figures 8.1, 8.2 and 8.3 it can be observed the mathematical approach hard caps the changeover from the unsaturated region to the saturated region. Thus, this model does fail to model the smooth transition obtained by simulation. It does, however, introduce very little error.

The results are in agreement with those compared to in section 6.6, as they should be, because the results of Garetto's model were used. The major contributions to the error are the transition and saturation regions. Again, the fewer the number of contention nodes, the larger the error becomes due to the failure of Garetto's modelling method.

Theoretically the approach makes sense, because the only change for the throughput is the time available to transmit packets. Decreasing the total time available to transmit packets will decrease the throughput accordingly.

Figure 8.2 illustrates the principle of a too small superframe. The average and relative percentage errors for this scenario increases because of the initial overshoot when the saturation throughput is reached. The results, do however, indicate the model to be a fair approximation to this behaviour.

8.1.4 PCF throughput scaling

The throughput of PCF was defined in section 7.3.1 and the same definition is used. Note again that λ_{Sat} is the packet arrival rate at which saturation occurs.

8.2 Combined PCF Throughput

The combined throughput of all nodes is obtained by simple superposition of the scaled throughput of DCF nodes operating in the CP and the throughput of PCF nodes.

8.2.1 Results and discussion

This method proves to be extremely accurate (see Figures 8.4 to 8.6) in determining the total throughput of a PCF network with nodes operating in both the CP and CFP regions. The results are in agreement with those obtained in section 6.6, in that the throughput is underestimated in the saturated region. They also agree with results obtained in section 7.3.2 with the throughput obtained for PCF. The results of Section 8.1 coincide with results here, and the major contributor to error is the method of scaling the DCF throughput.

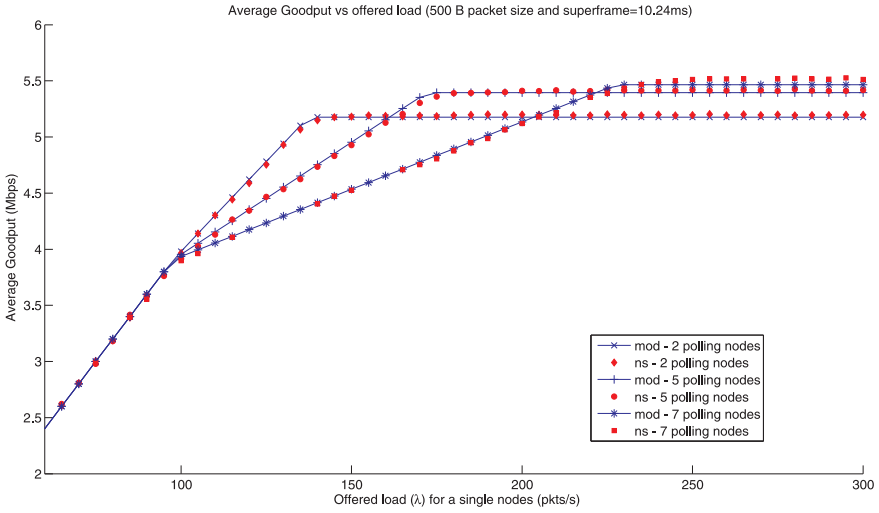


Figure 8.4: Total combined goodput versus the average packet arrival where the number of polling nodes is varied (10 nodes with 500B packet sizes and $T_{SF} = 10.24ms$)

Table 8.4: Combined throughput for 10 nodes with a variable number of CFP and CP nodes (500B packet size and $T_{SF} = 10.24ms$)

Number of CFP nodes	Average error (kbps)	Percentage error	Average saturation throughput simulation (Mbps)	Average saturation throughput model (Mbps)
2	17.093	0.40324	5.196	5.176
5	18.825	0.43836	5.418	5.395
7	28.064	0.61483	5.513	5.466

Table 8.5: Combined throughput for 10 nodes with a variable number of CFP and CP nodes (1000B packet size and $T_{SF} = 10.24ms$)

Number of CFP nodes	Average error (kbps)	Percentage error	Average saturation throughput simulation (Mbps)	Average saturation throughput model (Mbps)
2	151.92	2.5542	6.895	6.671
5	48.266	0.77228	7.028	6.977
7	34.791	0.57477	7.196	7.177

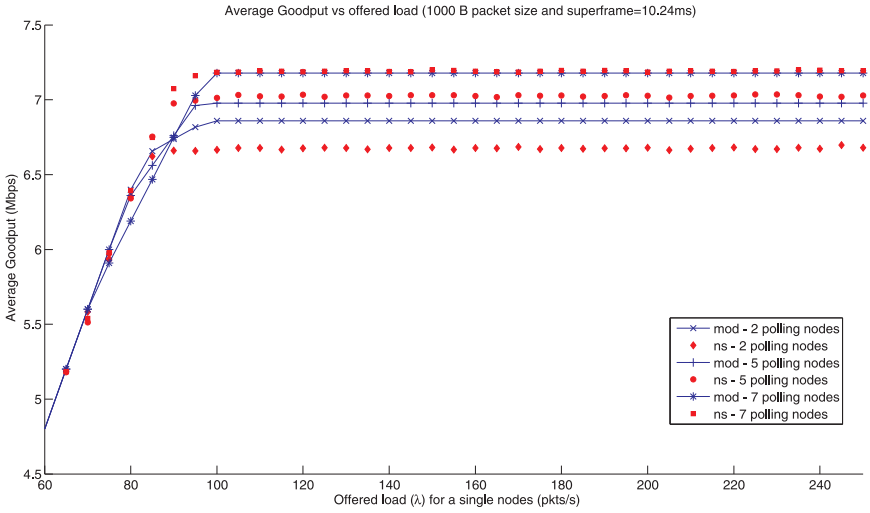


Figure 8.5: Total combined goodput versus the average packet arrival where the number of polling nodes is varied (10 nodes with 1000B packet sizes and $T_{SF} = 10.24ms$)

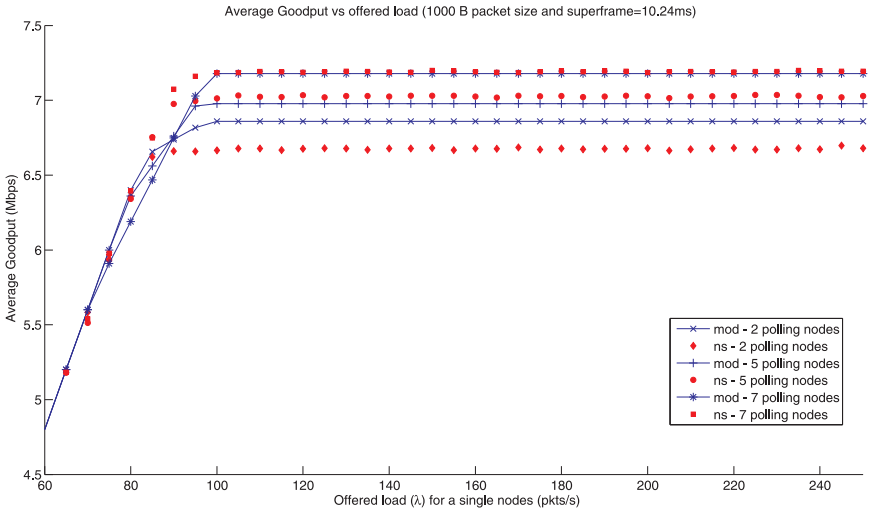


Figure 8.6: Total combined goodput versus the average packet arrival where the number of polling nodes is varied (10 nodes with 1000B packet sizes and $T_{SF} = 12.28ms$)

Table 8.6: Combined throughput for 10 nodes with a variable number of CFP and CP nodes (1000B packet size and $T_{SF} = 12.24ms$)

Number of CFP nodes	Average error (kbps)	Percentage error	Average saturation throughput simulation (Mbps)	Average saturation throughput model (Mbps)
2	149.31	2.5103	6.678	6.851
5	83.146	1.3304	7.03	6.928
7	83.58	1.3808	7.199	7.102

8.3 Scaling packet delay

In this section the same approach will be used as described in section 8.1, that is, to work with percentage fractions of the CFP and CP in order to scale the packet delay when nodes are operating in a superframe. The percentage fractions are determined by using the expected service cycle time $E[X_{SC}]$.

8.3.1 Scaling DCF packet delay

As discussed in section 6.6.1, we used the Chatzimisios model and included the empty buffer probability to compensate for non-saturated conditions. The delay for each DCF packet operating in the CFP will increase when a superframe is introduced, consuming some of the transmission time, especially for higher packet arrival rates. The packet arrival rate (offered load) at which saturation will take place will also decrease, because there is less transmission time for DCF nodes.

The method devised to deal with this problem for this project follows a similar approach to that with which the PCF packet delay was determined in section 7.4. The Chatzimisios delay is scaled according to the probability of certain events and the expected values of these events.

Figure 8.7 shows a schematic representation of a tagged packet arrival at time t_1. The arrival is at an empty queue and thus the packet will be delayed by the remainder of the CFP, which is t_x seconds. The packet would have been transmitted at T_1 seconds but, because of the additional delay, is now transmitted only at T_x. This occurrence will be referred to as case C1. Note, however, that the condition for which this is true will be that the packet delay is less than the duration of the CP (T_{CP}). This is a reasonable assumption, because an empty queue implies that we are operating under non-saturated conditions and that the packet delay will be smaller. Note, however, the possibility that if the duration of the superframe is short and the duration of the CFP consumes a high percentage of the superframe, which makes

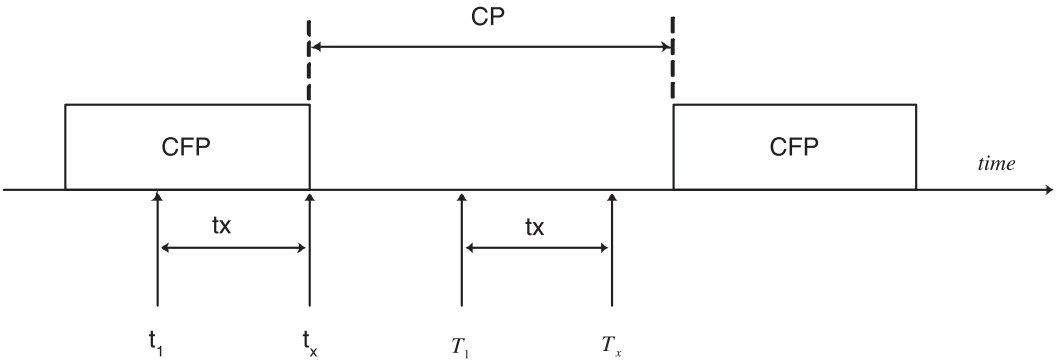


Figure 8.7: A schematic representation of a tagged packet arrival at an empty queue during a CFP

the duration of the CP too short, the packet will only be transmitted during the next CP. The assumption is that this does not happen.

The probability for case C1 of a packet arriving during a CFP is given as follows

$$P[C1] = \frac{T_{CFP}}{T_{SF}} \tag{8.3.1}$$

where T_{CFP} is the duration of the CFP and T_{SF} the duration of a superframe. If the assumption is that the packet arrival is uniformly distributed during a CFP, the packet arrival will be in the range $[0, T_{CFP}]$. T_{CFP} is determined by E[Xsc] and the expected value for the packet delay for C1 is thus $E[X_{SC}]/2$. The delay added to the Chatzimisios delay would be

$$E[X_{C1}] = P[C1] \cdot \frac{E[X_{SC}]}{2} = \frac{T_{CFP}}{T_{SF}} \cdot \frac{T_{CFP}}{2} = \frac{(E[X_{SC}])^2}{2 \cdot T_{SF}} \tag{8.3.2}$$

Figure 8.8 shows a schematic representation of a tagged packet arrival at time t_1. The time remaining in the current CP is not enough for the packet to be transmitted. When the CFP starts, the backoff timer will be paused, be restarted when the CFEnd is received, and the packet will only be transmitted during the next CP. This arrival will be delayed by the duration of the CFP, which is t_x seconds. The original transmission would have been transmitted at T_1 seconds but, because of this additional delay, it is now only transmitted at T_x seconds, which will be referred to as case C2. For this to be true, the condition is that the packet delay $E_C[D]$ is less than the remaining duration of the CP (T_{CP}). The probability of C2 is

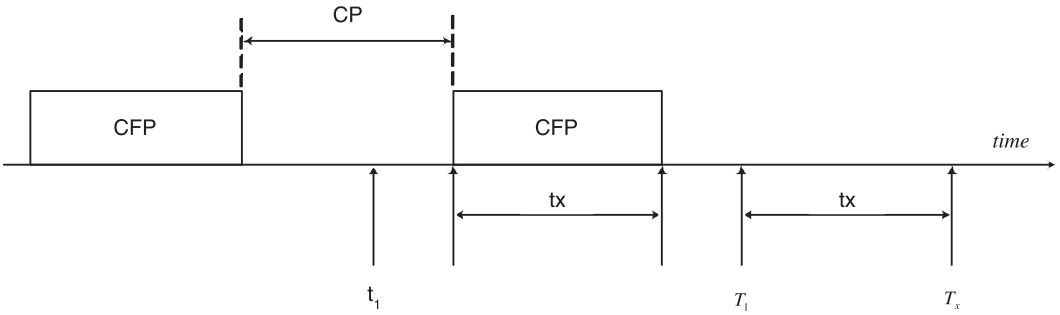


Figure 8.8: A schematic representation of a tagged packet arrival where not enough time remains for it to be serviced in the current CP and it will therefore only be serviced in the next CP.

$$P[C2] = \frac{E_C[D]}{T_{CP}} \tag{8.3.3}$$

where $E_C[D]$ is the average delay obtained by using the Chatzmisios method and T_{CP} the duration of the CP.

The expected delay added to each packet for case C2 is determined as

$$E[X_{C2}] = P[C2] \cdot T_{CFP} = \frac{E_C[D]}{T_{CP}} \cdot T_{CFP} \tag{8.3.4}$$

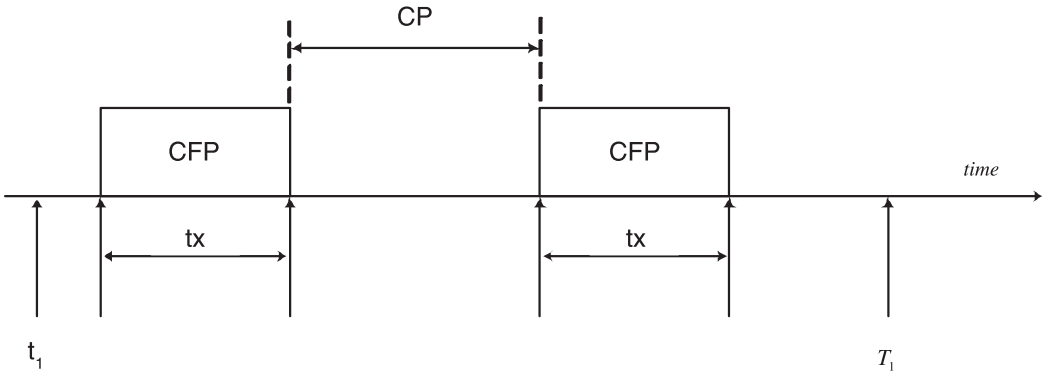


Figure 8.9: A schematic representation of a tagged packet arrival delayed by two CFPs.

Figure 8.9 shows a schematic representation of a tagged packet arrival at time $t1$. The time remaining in the current CP is not enough for the packet to be transmitted. The duration of $E_C[D]$ is also longer than T_{CP}, resulting in the packet being delayed

by two CFPs and this will be referred to as case C3. The condition to be satisfied for this to hold true is $E_C[D] - T_{CP} > 0$, and the probability for C3 is

$$P[C3] = \frac{E_C[D] - T_{CP}}{T_{CP}} \tag{8.3.5}$$

The added delay to $E_c[D]$ is given by

$$E[X_{C3}] = P[C3] \cdot 2 \cdot T_{CFP} = \frac{E_C[D] - T_{CP}}{T_{CP}} \cdot 2 \cdot T_{CFP} \tag{8.3.6}$$

The approach outlined above can be repeated for cases where $E_C[D]$ is larger than multiple CPs, for example if $E_C[D] > 2 \cdot T_{cp}$

If $T_{cp} < E_C[D] < 2 \cdot E_c[D]$, the added total delay would be

$$E_{cn}[D] = E_C[D] + E[X_{C1}] + E[X_{C2}] + E[X_{C3}]. \tag{8.3.7}$$

CFPs in a superframe structure will influence the relationship of the delay to the offered load. The approximation used to compensate for this was to scale the packet arrival rate with the following fraction

$$P_F[\lambda] = \lambda(T_{CP}/T_{SF}) \tag{8.3.8}$$

where λ is the packet arrival rate.

This is explained by Little's law, which states $N = \lambda T$. Increasing the time the packet spends in the system increases the number of packets in the queue. To scale the buffer occupancy in accordance with the packet delay, the packet arrival rate has to be scaled.

8.3.2 PCF packet delay scaling

Please refer to section 7.4 for a thorough discussion of the packet delay for PCF nodes.

8.3.3 Results

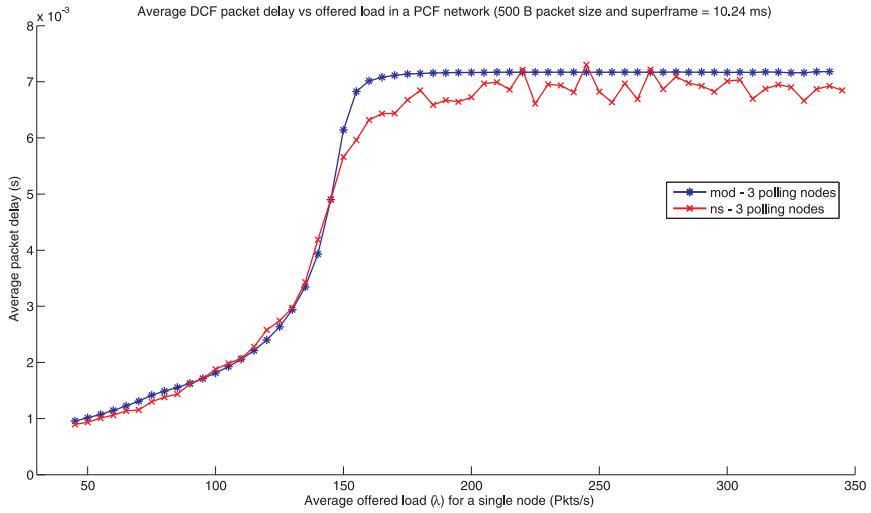


Figure 8.10: Scaled average DCF delay for 3 pollable nodes (500B packets and superframe=10.24ms)

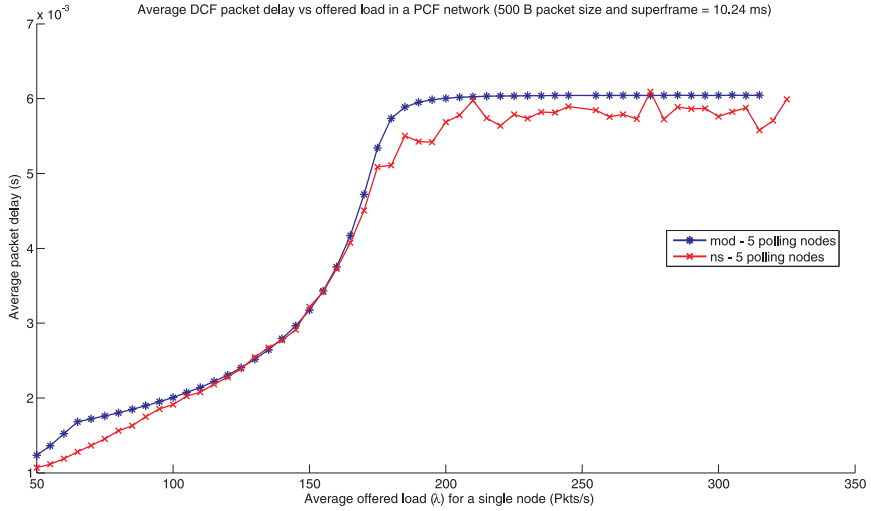


Figure 8.11: Scaled average DCF delay for 5 pollable nodes (500B packets and superframe=10.24ms)

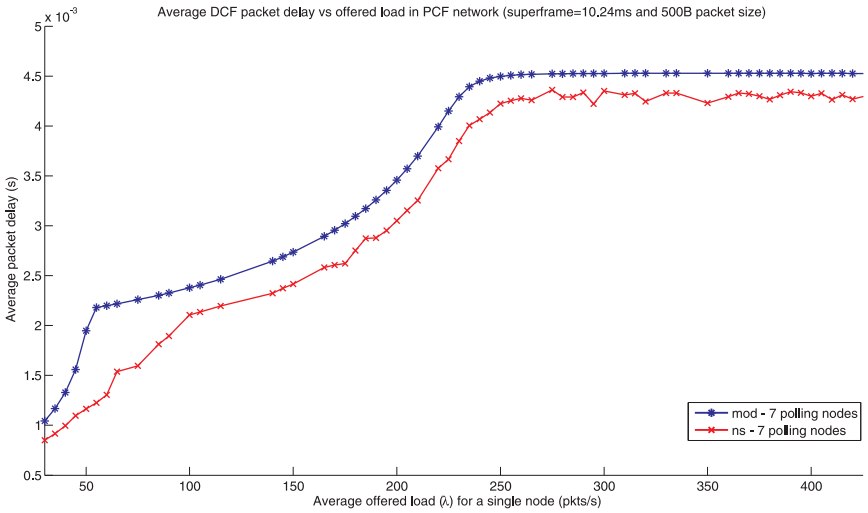


Figure 8.12: Scaled average DCF delay for 7 pollable nodes (500B packets and superframe=10.24ms)

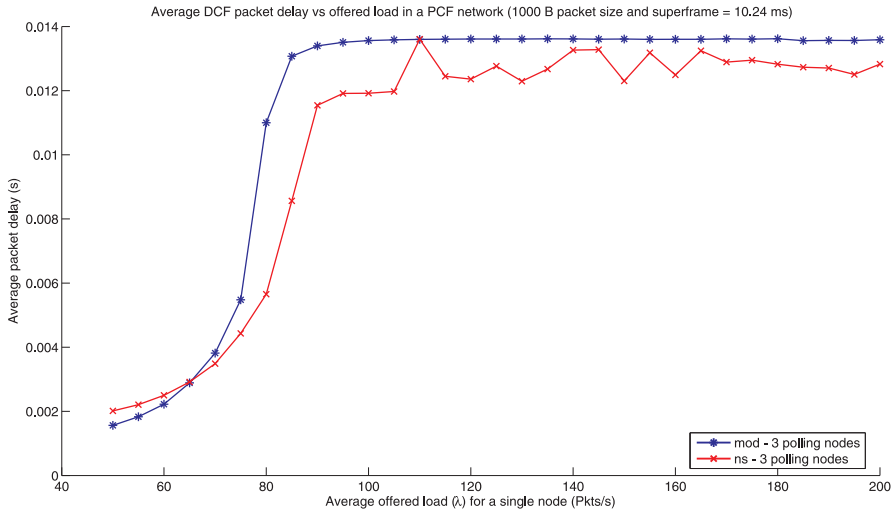


Figure 8.13: Scaled average DCF delay for 3 pollable nodes (1000B packets and superframe=10.24ms)

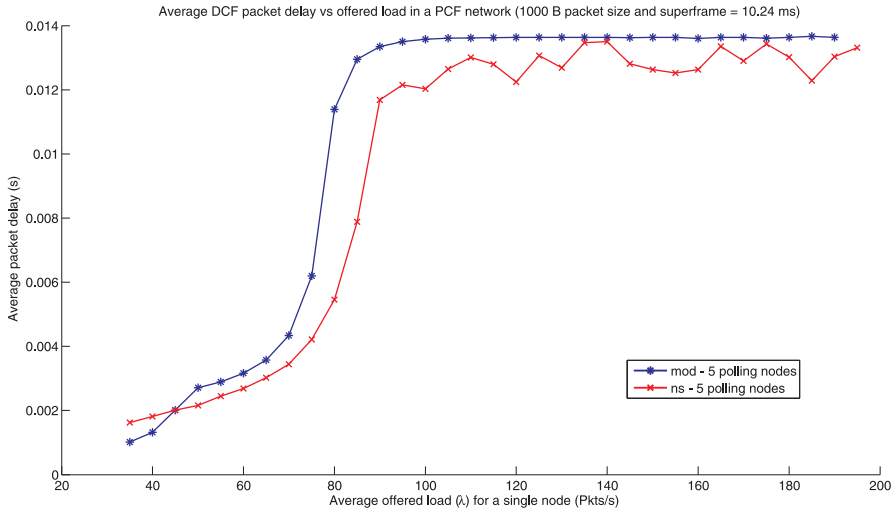


Figure 8.14: Scaled average DCF delay for 5 pollable nodes (1000B packets and superframe=10.24ms)

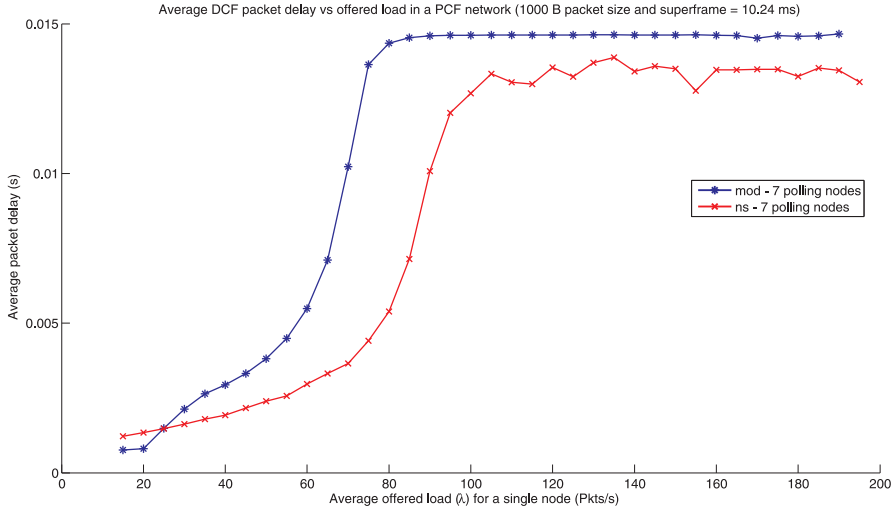


Figure 8.15: Scaled average DCF delay for 7 pollable nodes (1000B packets and superframe=10.24ms)

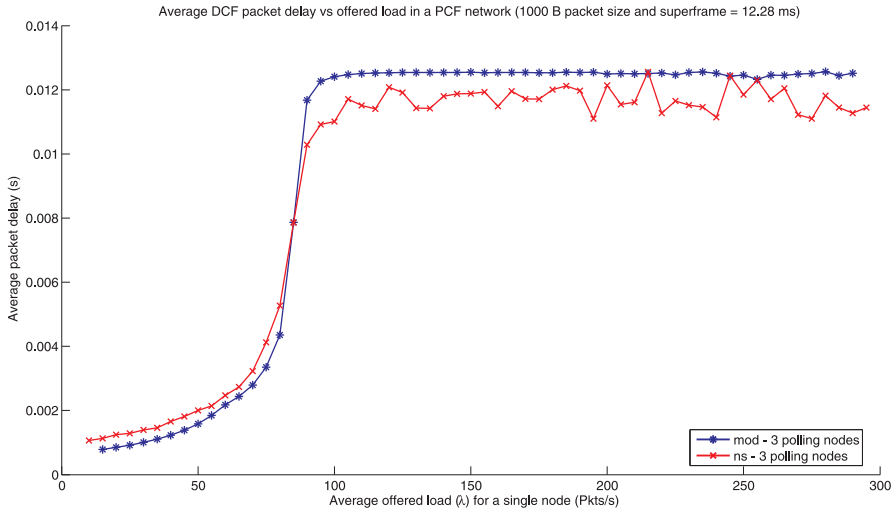


Figure 8.16: Scaled average DCF delay for 3 pollable nodes (1000B packets and superframe=12.28ms)

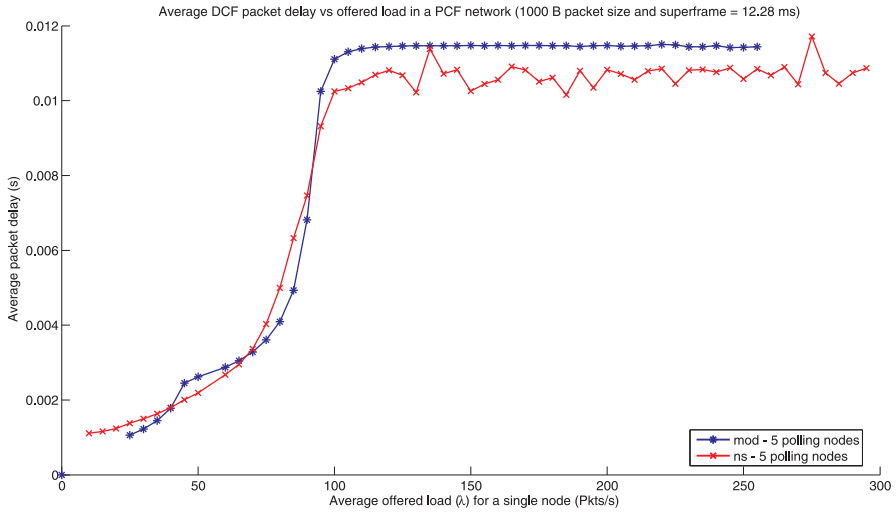


Figure 8.17: Scaled average DCF delay for 5 pollable nodes (1000B packets and superframe=12.28ms)

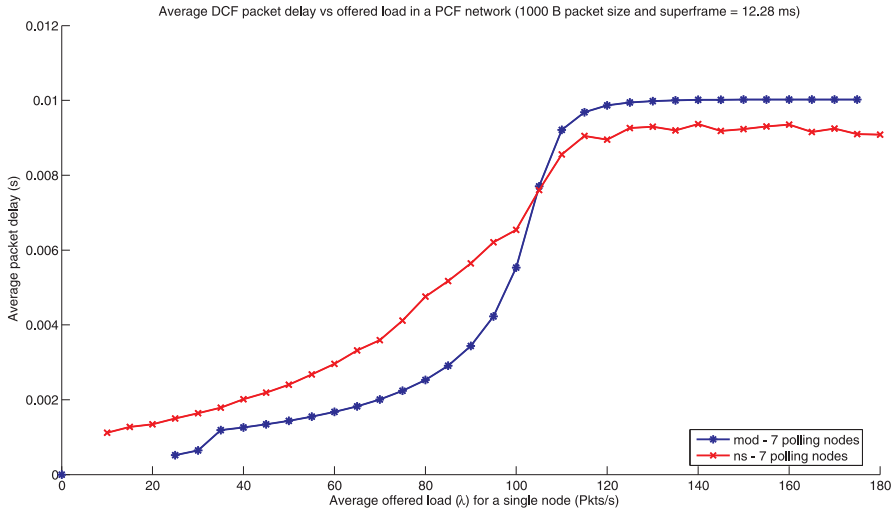


Figure 8.18: Scaled average DCF delay for 7 pollable nodes (1000B packets and superframe=12.28ms)

Table 8.7: Summary of results for figures 8.10, 8.11 and 8.12 with 500B packets

Number of CP nodes	Average error (ms)	Percentage error	Average saturation delay simulation (ms)	Average saturation delay model (ms)
3	0.42351	13.771	4.334	4.527
5	0.36323	7.8323	5.888	6.043
7	0.25905	5.0598	6.966	7.171

Table 8.8: Summary of results for figures 8.13, 8.14 and 8.15 with 1000B packets

Number of CP nodes	Average error (ms)	Percentage error	Average saturation delay simulation (ms)	Average saturation delay model (ms)
3	0.44864	22.267	12.83	13.31
5	0.6078	37.382	12.69	13.63
7	0.80107	56.613	13.46	14.27

Table 8.9: Summary of results for figures 8.16, 8.17 and 8.18 with 1000B packets

Number of CP nodes	Average error (ms)	Percentage error	Average saturation delay simulation (ms)	Average saturation delay model (ms)
3	1.06	17.085	9.301	10.02
5	0.69113	8.1034	10.91	11.47
7	0.74622	7.7018	11.95	12.53

Table 8.10: Service cycle times for a PCF network of 10 nodes (500B packets and Tsf=10.24ms)s

Number of CP nodes	Service cycle time ns (ms)	Service cycle time mathematical model (ms)	Average remaining contention period time mathematical model(ms)	Normal average DCF saturation delay ns2 (ms)	Normal average DCF saturation delay mathematical model (ms)
3	4.7838	4.8168	5.4232	2.242	2.447
5	3.4322	3.464	6.776	4.045	3.792
7	2.0801	2.1112	8.1288	5.306	5.724

Table 8.11: Service cycle times for a PCF network of 10 nodes (1000B packets and Tsf=10.24ms)

Number of CP nodes	Service cycle time ns (ms)	Service cycle time mathematical model (ms)	Average remaining contention period time mathematical model(ms)	Normal average DCF saturation delay ns2 (ms)	Normal average DCF saturation delay mathematical model (ms)
3	7.462	7.589	2.651	3.473	3.787
5	5.407	5.444	4.796	5.913	6.385
7	3.268	3.2992	6.9408	8.35152	9.082

Table 8.12: Service cycle times for a PCF network of 10 nodes (1000B packets and Tsf=12.28ms)

Number of CP nodes	Service cycle time ns (ms)	Service cycle time mathematical model (ms)	Average remaining contention period time mathematical model(ms)	Normal average DCF saturation delay ns2 (ms)	Normal average DCF saturation delay mathematical model (ms)
3	7.462	7.589	4.691	3.473	3.787
5	5.407	5.444	6.836	5.913	6.385
7	3.268	3.2992	8.9808	8.35152	9.082

8.3.4 Discussion

The results are consistent with those obtained in section 6.6.1 As can be seen from the previous section, our model estimates the scaled DCF delay with some degree of error, especially for larger packet sizes. Unfortunately this method of delay scaling is limited by the results obtained by Garetto's model in conjunction with the Chatzimisios method (refer to sections 6.6.1, 6.6.1.6 and 6.6.2). The conditional collision probability is overestimated by Garetto's model and using this with Chatzimisios method will result in the overestimation of the packet delay. The number of DCF nodes has a significant effect on the result obtained; the fewer the nodes, the larger the error becomes, whereas the larger the number of DCF nodes, the closer the approximation becomes.

From Table 8.10, the average packet delay is less than the contention period duration, implying that if the packet at the head of the buffer is dequeued with enough time remaining it might still transmit during the current CP and not be delayed by a CFP. Whereas in the case for the 1000B packet and 10.24ms superframe size, the average packet delay is larger than the average CP, therefore it is guaranteed that every packet will be delayed by at least one CFP. For example, for 3 polling and 7 contention period nodes with a $1000B$ packet size, (the last row of Table 8.11), the normal average saturation delay by the model is $9.082ms$, whereas the delay for the scaled model is $14.27ms$ (last row of Table 8.8). Note that as the normal delay is roughly one and half times larger than the expected contention period of $6.776ms$, the scaled delay will be increased by roughly one and a half times the duration of the CFP.

The steep rise at the knee of the curve, eventually smoothing out, especially for the $1000B$ and 10.24 ms superframe scaled packet delay, is caused by the T_{CP} and T_{CFP} scaling. The percentage fraction T_{CFP} increases, causing the initial steep rise, until the maximum service cycle time is reached, which causes the curve to smooth out.

In the unsaturated region, especially for Figures 8.13, 8.14 and 8.15, it can be noted that if the service cycle increases to become a significant portion of the superframe, and the number of contention nodes decreases, an overestimation of the packet delay is caused. Because the empty queue probability is overestimated, a packet which would have arrived at an empty queue, can now arrive at an empty queue during the CFP. When the CP restarts, the packet is ready to be transmitted and the empty queue probability is actually decreased, as it is measured only during the CP. This model has not completely compensated for this, but there is little effect on the overall results.

Comparing the results for the 1000B and 10.24ms superframe size to that for the 1000B and 12.28ms superframe size, it can be noticed that the model is better suited

to approximate the scaled packet delay, which is only delayed by one CFP. That is, for the case when there is roughly enough time in the CP for every DCF node to transmit at least once and that, therefore, the normal average packet delay is less than the duration of the CP.

8.4 Scaling of buffer occupancy

Only the scaling of the packet arrival rate is required to obtain the new buffer occupancy. As discussed in section 8.3.1, the average buffer occupancy is related by Little's law to the packet arrival rate multiplied by the average packet delay. In the previous section it was shown that, by introducing a superframe structure, the average packet delay is increased and that, therefore, the packet arrival rate decreases. The formula for the scaling of the packet delay is

$$P_F[\lambda] = \lambda(T_{CP}/T_{SF}) \tag{8.4.1}$$

where λ is the average packet arrival rate, T_{CP} the expected service cycle time and T_{SF} the duration of a superframe.

8.4.1 Results

Table 8.13: Summary of results for Figure 8.19 with 500B packets and superframe=10.24ms

Number of CP nodes	Average error (packets)	Percentage error (%)
3	0.72308	9.9455
5	0.6818	8.7784
7	0.53619	7.5076

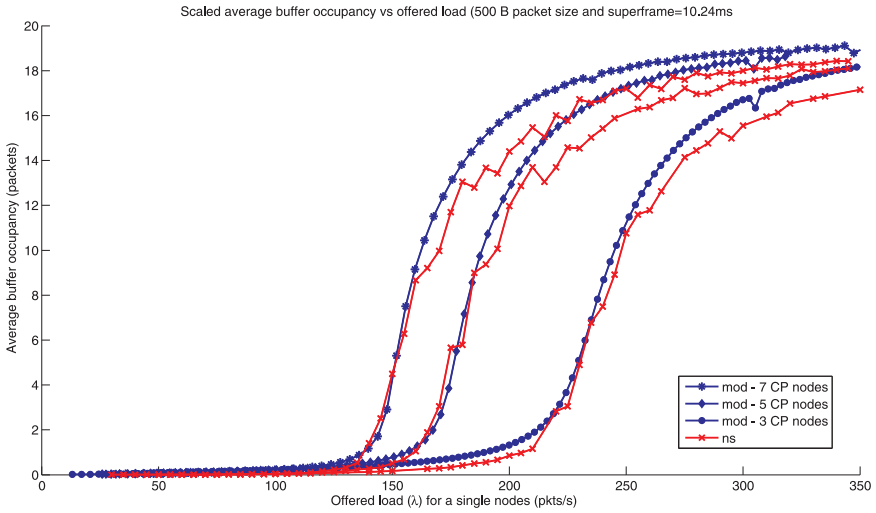


Figure 8.19: Scaled average DCF buffer occupancy versus total offered load comparison of mathematical modelling and simulation for 10 nodes and a variable number of contention nodes (500B packets and superframe=10.24ms)

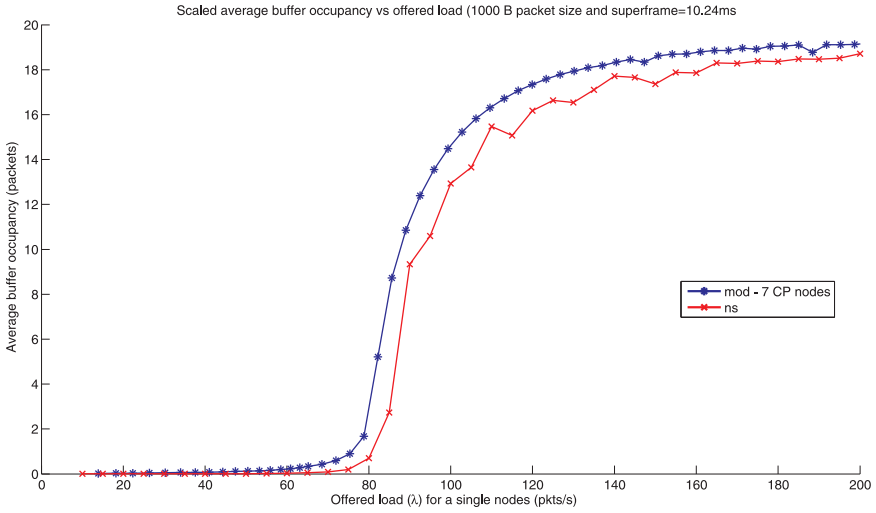


Figure 8.20: Scaled average DCF buffer occupancy versus total offered load comparison of mathematical modelling and simulation for 10 nodes, of which 7 are contention and 3 polling nodes (1000B packets and superframe=10.24ms)

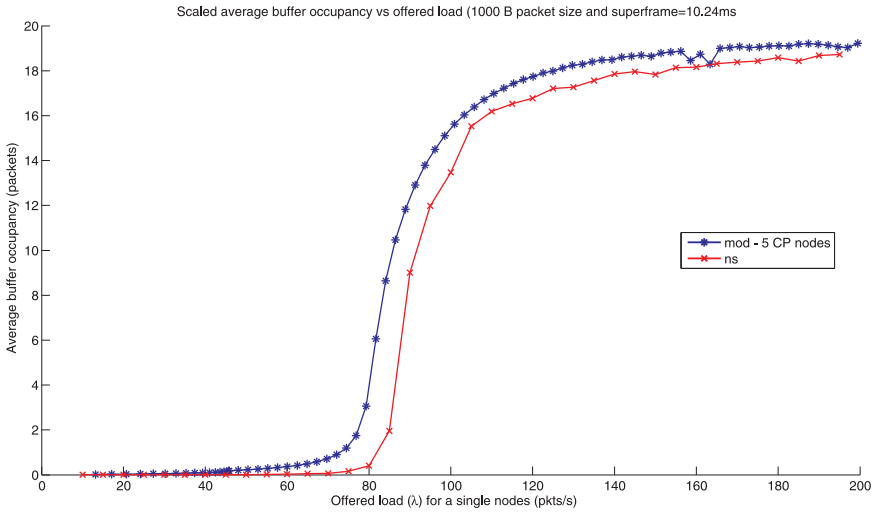


Figure 8.21: Scaled average DCF buffer occupancy versus total offered load comparison of mathematical modelling and simulation for 10 nodes, of which 5 are contention and 5 polling nodes (1000B packets and superframe=10.24ms)

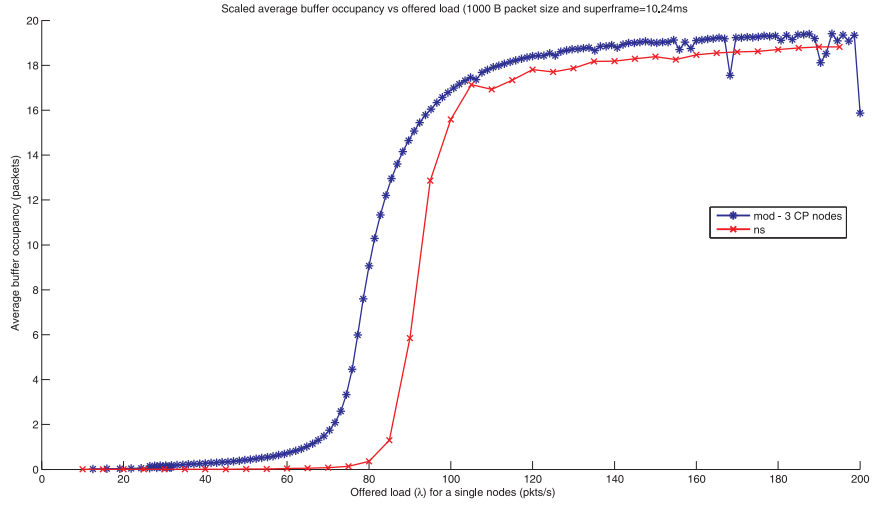


Figure 8.22: Scaled average DCF buffer occupancy versus total offered load comparison of mathematical modelling and simulation for 10 nodes, of which 3 are contention and 7 polling nodes (1000B packets and superframe=10.24ms)

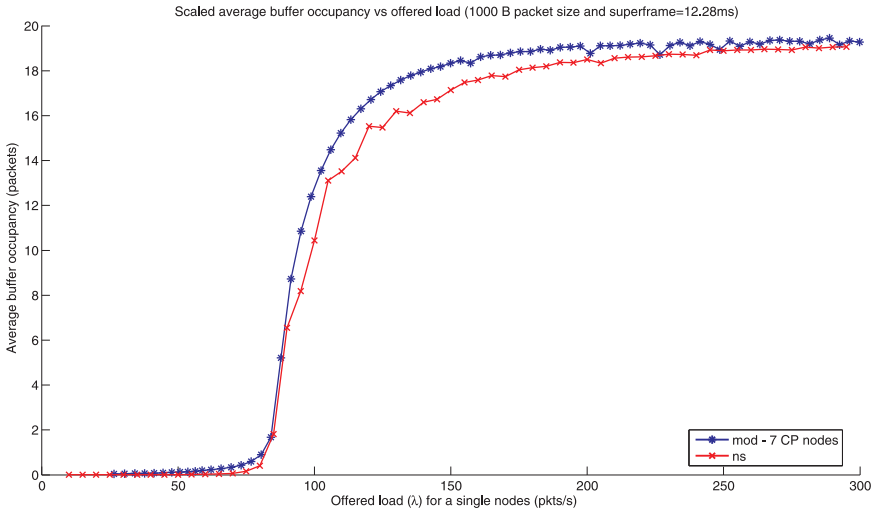


Figure 8.23: Scaled average DCF buffer occupancy versus total offered load comparison of mathematical modelling and simulation for 10 nodes, of which 7 are contention and 3 polling nodes (1000B packets and superframe=12.28ms)

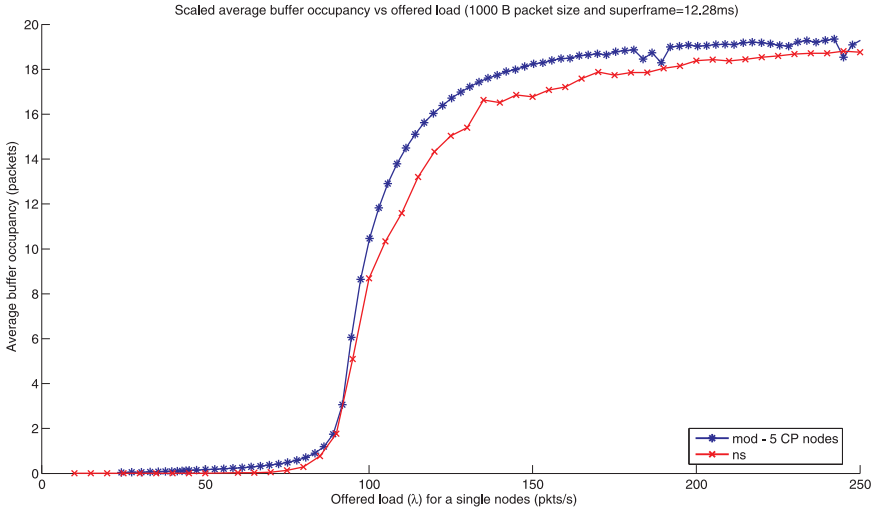


Figure 8.24: Scaled average DCF buffer occupancy versus total offered load comparison of mathematical modelling and simulation for 10 nodes, of which 5 are contention and 5 polling nodes (1000B packets and superframe=12.28ms)

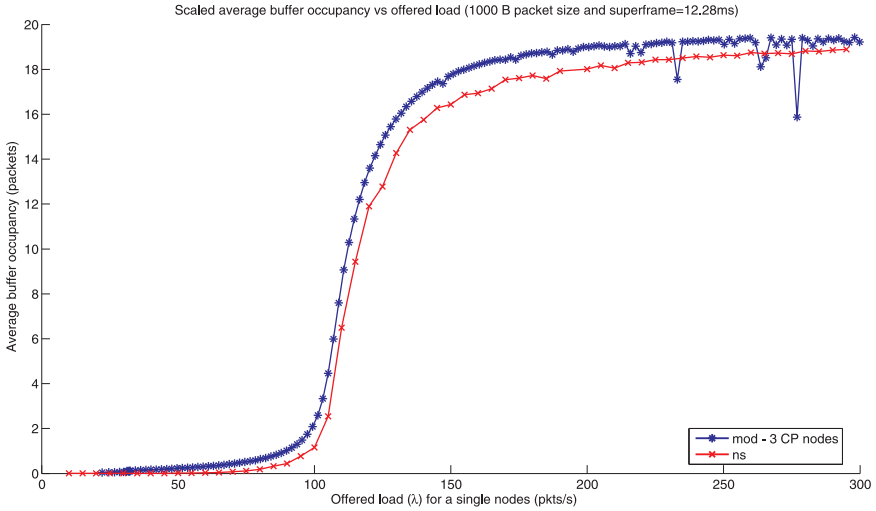


Figure 8.25: Scaled average DCF buffer occupancy versus total offered load comparison of mathematical modelling and simulation for 10 nodes, of which 3 are contention and 7 polling nodes (1000B packets and superframe=12.28ms)

Table 8.14: Summary of results for Figures 8.20, 8.21 and 8.22 with 1000B packets and superframe=10.24ms

Number of CP nodes	Average error (packets)	Percentage error (%)
3	1.4982	14.864
5	0.96633	9.7996
7	0.85181	7.8091

Table 8.15: Summary of results for Figures 8.23, 8.24 and 8.25 with 1000B packets and superframe=12.28ms

Number of CP nodes	Average error (packets)	Percentage error (%)
3	0.8037	6.9196
5	0.81732	6.883
7	0.71597	5.4132

8.4.2 Discussion

From the results (see Figure 8.19 to 8.25)it may be concluded that this approach serves as a fair approximation. As with the scaled average packet delay, the larger the percentage of time the service cycle time consumes of the superframe, and the less the number of contention nodes (with the value determined by Garetto's model), the larger the error becomes.

A higher packet arrival rate in the saturated region will result in the convergence of the buffer occupancies of the simulated and mathematical models thereby reducing the error obtained. Therefore, the measure of error will be increased with fewer nodes because the transition region between the saturated and unsaturated regions continues longer, increasing the error. In section 6.7.1 it was shown that Garetto's model does model fewer nodes as giving greater error, which is in agreement with the results obtained.

In the unsaturated region there are, on average, two or fewer packets in the buffer which will be overestimated by this approach, because the buffer values do not consider the effect the CFP has on the probability of an empty buffer. However, the probability exists that for these reasons there might be no packet arrival during a CFP, or that the buffer is empty. For the buffer occupancy will have to be scaled by less than it would be by T_{CFP}. The approach does, however, prove accurate for the saturated region.

Comparing the results for the 1000B and 10.24ms superframe size (see Figures 8.20 to 8.22) to that for the 1000B and 12.28ms superframe size (see Figures 8.23 to 8.25), it can be noticed that the model is better suited to approximate the scaled DCF buffer occupancy, where there is enough time for DCF nodes to tranmit in the CP. The results are consistent with those in section 8.3.3, and should be, because the buffer occupancy and packet delay is related by Little's law.

8.4.3 Summary

This chapter has discussed the scaling of the DCF throughput and packet delay when it works in conjunction with a superframe. It was shown that these values were determined with some degree of error, which can be attributed to the shortcomings of Garetto's model and omitting the scaling of the empty queue probability. The combined throughput of both contention and polling nodes operating together in a superframe was also covered. The approach in this case proved very accurate.

Chapter 9

Proposed protocol

In this chapter a proposed protocol to enable dynamic adaption between DCF and PCF modes of operation will be discussed. This will investigate which protocol, PCF or DCF, is better suited to which circumstance regarding throughput, delay and buffer occupancy. Also, a proposed protocol will be given, based on the investigation of the simulated and predicted results.

9.1 DCF and PCF tradeoffs

9.1.1 Motivation for study

As might have been noticed from the previous sections, PCF and DCF each have their strengths and weaknesses. To the authors knowledge no protocol has taken advantage of this fact to create an adaptive protocol. Bandwidth is a scarce commodity especially for wireless technology, and it is the goal of this project to optimise its use with existing technology.

9.2 PCF versus DCF trade-off study

9.2.1 Experimental setup

The experimental setup is the same as that outlined in Chapter 5 and the mathematical model as that in Chapter 8 .

9.2.2 Results and discussion

Investigating Figures 9.1 and 9.2, it becomes clear that in the unsaturated region DCF is the better protocol to use. The reasoning behind this is that PCF adds overhead by transmitting beacon and CFEnd frames, reducing the available time for data transmission; which results in a reduction in throughput. Also, nodes that are polled with no data to transmit consume valuable transmission time.

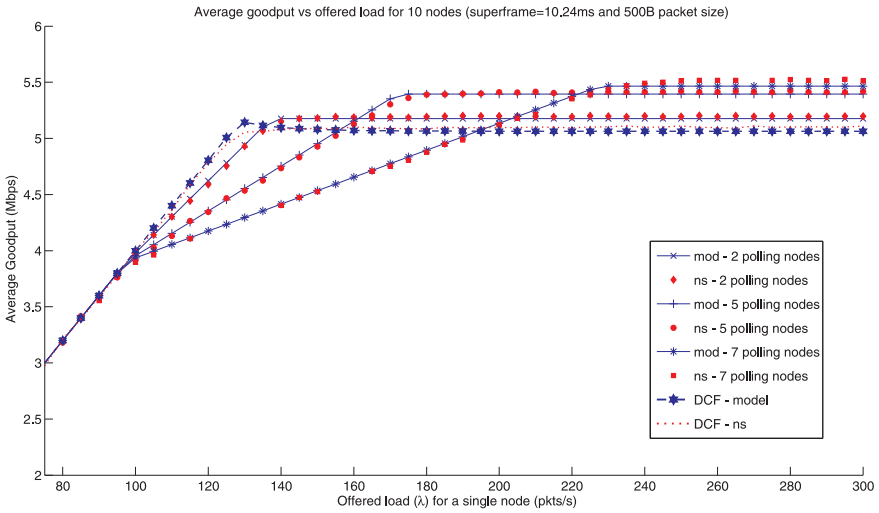


Figure 9.1: Comparison of average goodput vs offered load for various PCF configurations and DCF (for 10 nodes, constant 500B packet size and Tsf=10.24ms)

Table 9.1: Result summary for Figure 9.1 for a constant 500B packet size

Number of CFP nodes	Average saturation delay simulation (ms)	Average saturation delay model (ms)	Scaled average saturation delay simulation (ms)	Scaled average saturation delay model (ms)
2	6.1682	6.577	7.396	7.689
5	3.792	4.045	5.888	6.043
7	2.236	2.4464	4.329	4.527
10 DCF nodes (no superframe)	7.926	8.339		

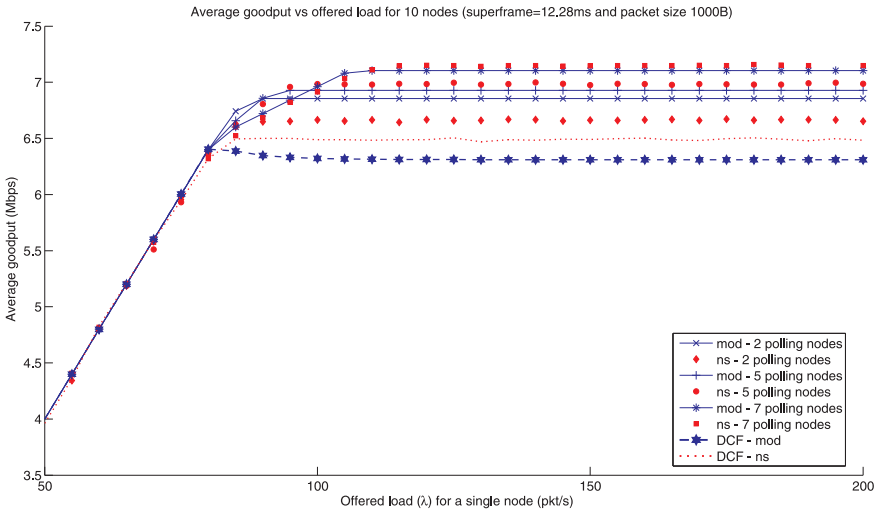


Figure 9.2: Comparison of average goodput vs offered load for various PCF configurations and DCF (for 10 nodes, constant 1000B packet size and Tsf=12.28ms)

Table 9.2: Result summary for Figures 9.2 for a constant 1000B packet size

Number of CFP nodes	Average saturation delay simulation (ms)	Average saturation delay model (ms)	Scaled average saturation delay simulation (ms)	Scaled average saturation delay model (ms)
2	9.3715	9.8832	12.02	12.14
5	5.913	6.3665	10.83	11.47
7	3.473	3.787	10.44	9.914
10 DCF nodes (no superframe)	12.34	13.39		

The packet delay is less in the unsaturated region for the DCF mode, because of the overhead that is added to the packet delay by CFPs, and PCF nodes that are serviced at fixed intervals; which results in a larger packet delay. This was shown in Chapter 8 and will also be proved here in a subsequent section.

In the saturated region, PCF is the better mode of operation, as it delivers a better throughput due to fewer collisions. In addition, in the saturated region PCF has an almost constant packet delay due to the superframe repetition interval, whereas DCFs delay increases even further resulting in a larger delay than PCF nodes.

The results obtained from mathematical modelling are in agreement with the result obtained by simulation (see Chapter 8).

9.3 Proposed protocol

From our simulations and mathematical modelling it was noticed that as soon as the DCF throughput started saturating, the buffer occupancy started increasing and the packet delay reached its maximum value. From the data points indicated on figure 9.4 it can be observed for example for 15 DCF nodes that there is a drastic increase in the average number of packets in the buffer between the average packet arrival rates of 80 and 105 packets per second. From the data points on figure 9.3 between the packet arrival rates of 80 and 105 packets per second, it can be observed that this is the region where there is a decrease in the rate at which the throughput relates to the packet arrival rate. At roughly 80 packets per second the rate of change is not linear anymore between the throughput and the packet arrival rate and this is when the buffer occupancy shows a significant rate of increase. At roughly 105 packets per second the throughput has reached its saturation value, and the packet arrival rate at which saturation occurs is when the buffer occupancy reaches half of its maximum buffer occupancy value which is for this specific case 10 packets.

The packet delay shows the same characteristics as explained for 15 DCF nodes between 80 and a 105 packets per second, and should, because it relates to the buffer occupancy by Little's law. Please refer to figures 6.9 to 6.11, and 6.18 to 6.21 for further examples of this. This characteristic was used to create the proposed protocol by monitoring the queue length, to determine when DCF nodes enter the saturated region and to switch over to PCF.

It is also possible to use the throughput or packet delay to create this protocol but a lookup table would then be needed to determine when the PCF mode should be

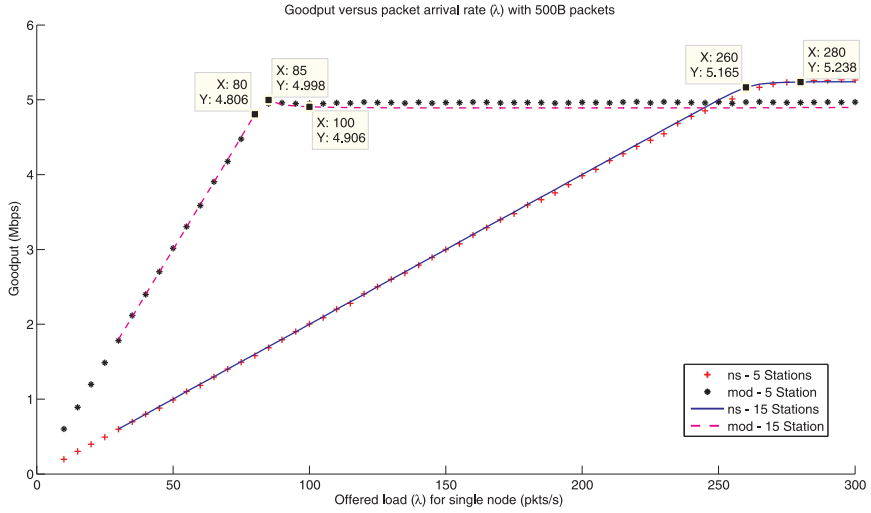


Figure 9.3: Average goodput vs offered load for DCF 5 and 15 nodes with a constant 500B packet size (data points are included on the figure)

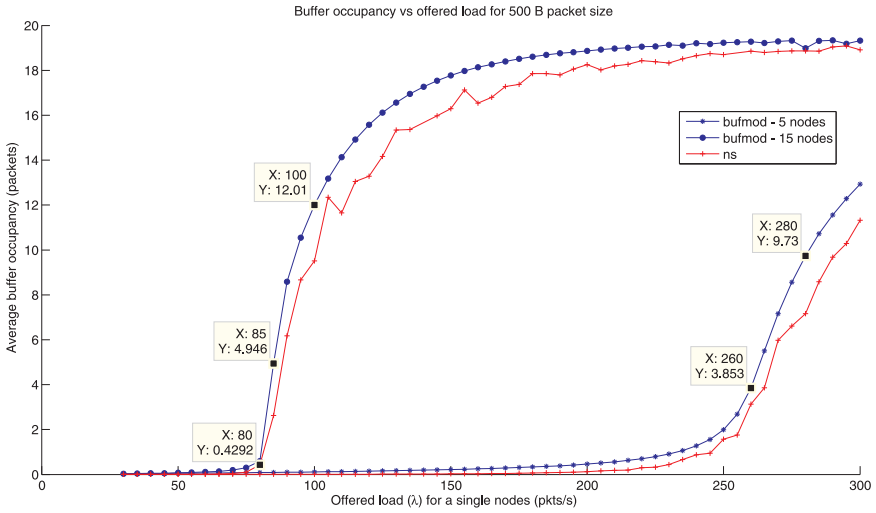


Figure 9.4: Average buffer occupancy vs offered load for DCF 5 and 15 nodes with a constant 500B packet size (data points are included on the figure)

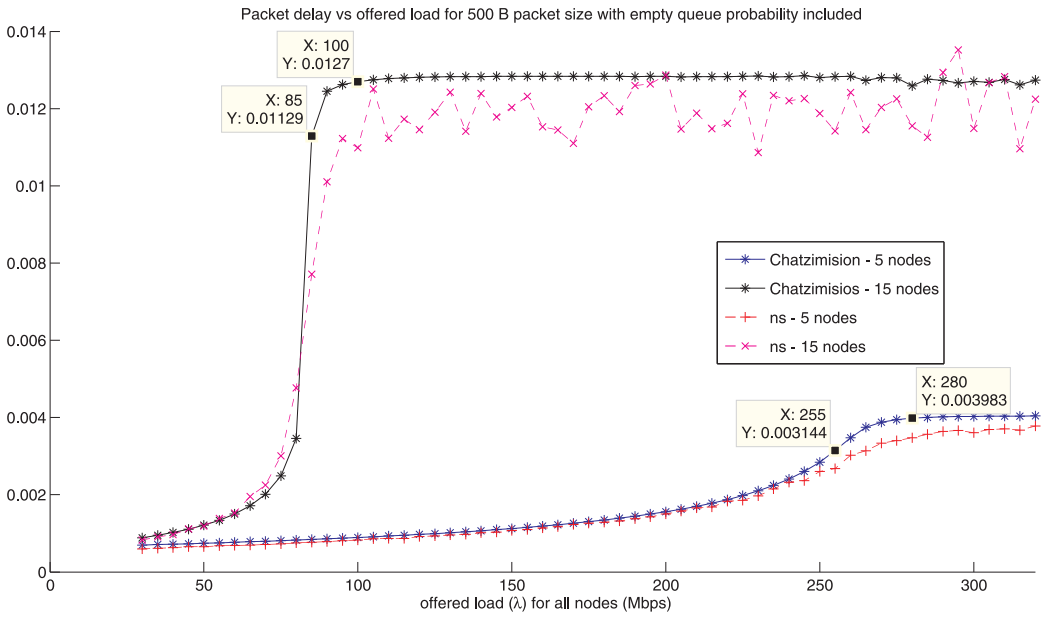


Figure 9.5: Average packet delay vs offered load for DCF 5 and 15 nodes with a constant 500B packet size (data points are included on the figure)

started and which nodes to add to the polling list. This is a tedious approach.

9.3.1 Proposed protocol design

The general overview for the design of the protocol is that every station monitors its queue length and adds a single bit in its header that is transmitted with every transmission. If the queue length exceeds a predefined length the bit is set to one, otherwise the value is zero (see section 3.10.1).

With the reception of each transmitted packet the base station monitors this single bit. If there are no nodes on the polling list and the bit's value is one, the station is added to the polling list and the beacon timer is started. If there are already nodes on the polling list, the node is added to the polling list and the beacon timer resumes as normal. At the expiration of the beacon timer the beacon is transmitted and the CFP is started. Figure 9.6 illustrates the flow of the instructions that takes place for the proposed protocol.

There is a problem that arises using this approach, which is that pollable stations should only transmit during the CFP. To ensure this, with every CFEnd transmitted there is also a single bit that indicates by having a value of one, another CFP is to come. If this bit is zero, all stations know to operate only using DCF. Each station is aware of the previous transmission it had and whether the queue length bit was

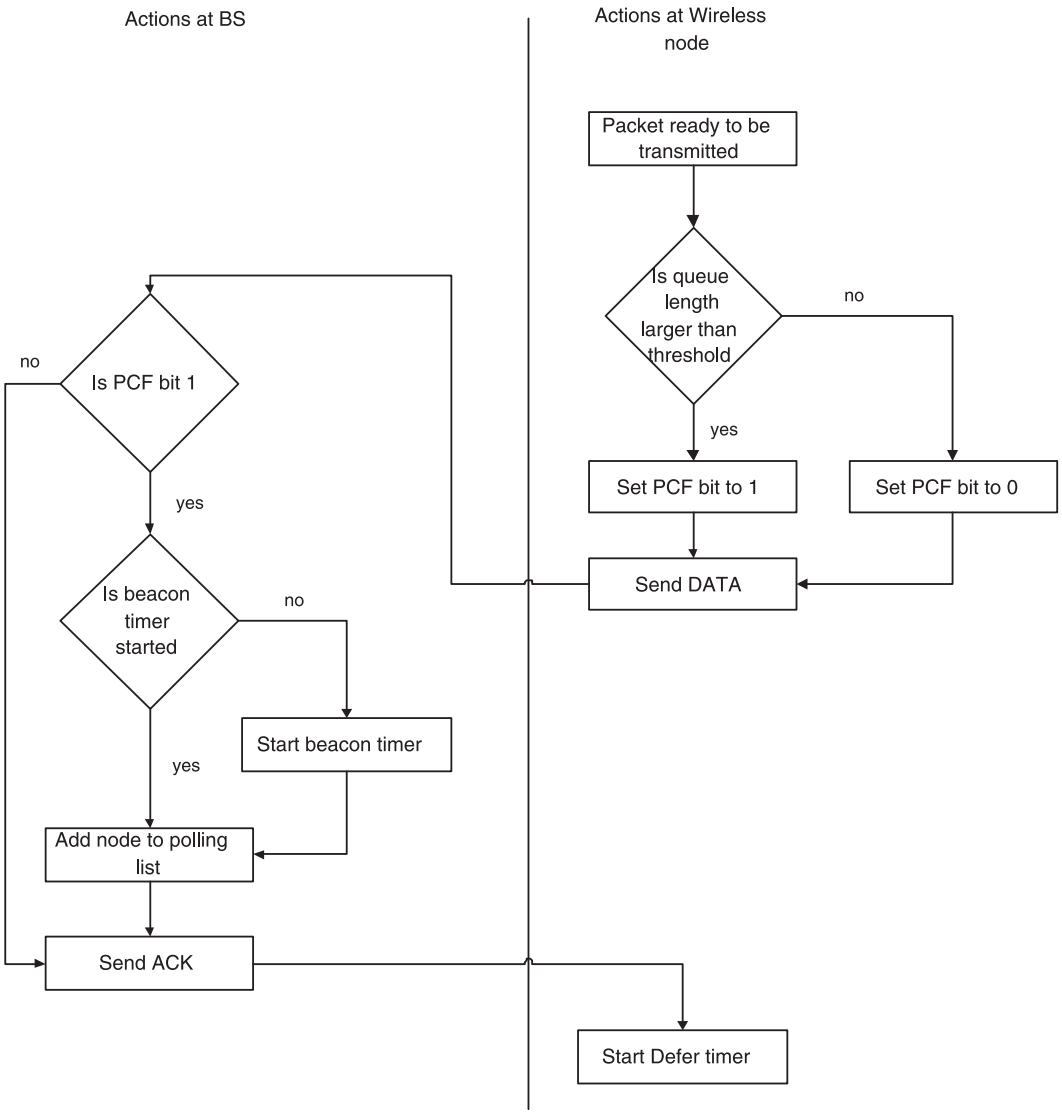


Figure 9.6: Illustration of the flow of instructions for the proposed protocol

set. With the knowledge of expecting another CFP, these stations will not transmit during the CP. The base station knows if another beacon is to come by checking to see if there are one or more nodes on the polling list.

Predicting and proving the behaviour of the proposed protocol, which is dependant on the buffer occupancy, led to two approaches being used. The first has a total number of N nodes, of which x have a low fixed and y a high packet arrival rate. In the second approach, all nodes are operating at the same rate. The first approach will be illustrated only by simulation, because of the mathematical limitations created by nodes operating at multiple rates and queueing modelling (this will result in G/G/1 queueing model that cannot be solved). The second approach will be proved by simulation and mathematical modelling.

9.3.2 Proposed protocol assumptions

The main assumptions and configuration parameters are as follows:

- A fixed number of competing stations are considered accessing the same wireless channel.
- There are no hidden terminals.
- Stations are equally likely to access the channel.
- The communication channel is error-free.
- The superframe value is large enough that the CFP does not saturate before DCF nodes reach their saturation throughput.
- All packets have a fixed size
- The same experimental procedures are used as outlined in Chapter 5

9.3.3 Proposed protocol simulation results

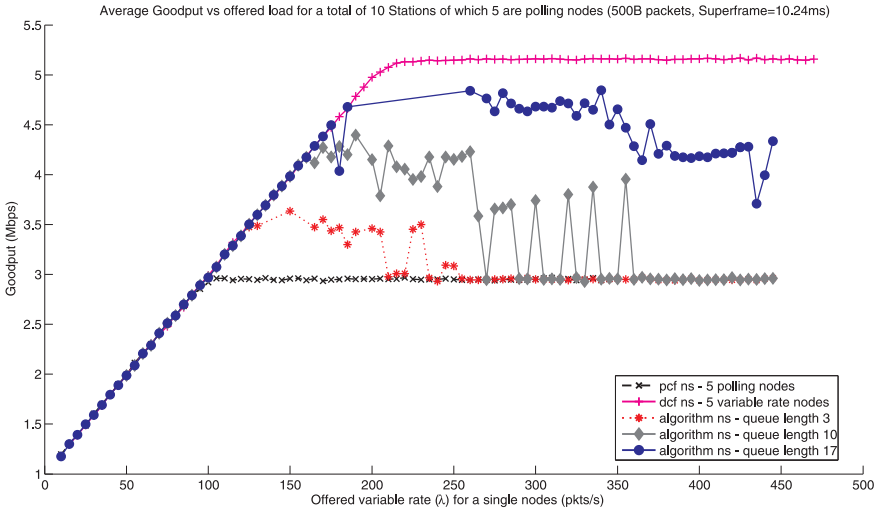


Figure 9.7: Average Goodput versus the packet arrival rate for a single node where the queue length parameter for the proposed protocol is varied (10 nodes, of which 5 have a fixed rate of $\lambda_{slow} = 50$ and 500B packet size)

Table 9.3: Result summary for Figure 9.9

Result type	Result
Average DCF saturation goodput value	6.522 (Mbps)
Average PCF saturation goodput value	6.64 (Mbps)
Average goodput for algorithm with queue length parameter 10	6.659 (Mbps)
Average goodput for algorithm with queue length parameter 15	6.676 (Mbps)
Packet arrival rate at which saturation starts for DCF	100 (pkts/s)
Packet arrival rate at which saturation starts for PCF	100 (pkts/s)
Packet arrival rate at which maximum goodput value occurs for queue length parameter 10	115 (pkts/s)
Packet arrival rate at which maximum goodput value occurs for queue length parameter 15	135 (pkts/s)
Max goodput value for queue length parameter 10	6.863 (Mbps)
Max goodput value for queue length parameter 15	6.964 (Mbps)

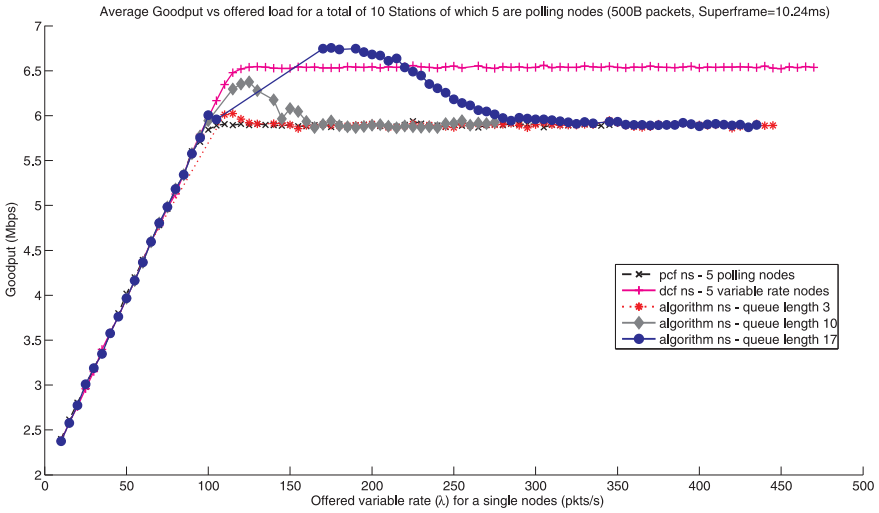


Figure 9.8: Average Goodput versus the packet arrival rate for a single node where the queue length parameter for the proposed protocol is varied (10 nodes, of which 5 have a fixed rate of $\lambda_{slow} = 50$ and 1000B packet size)

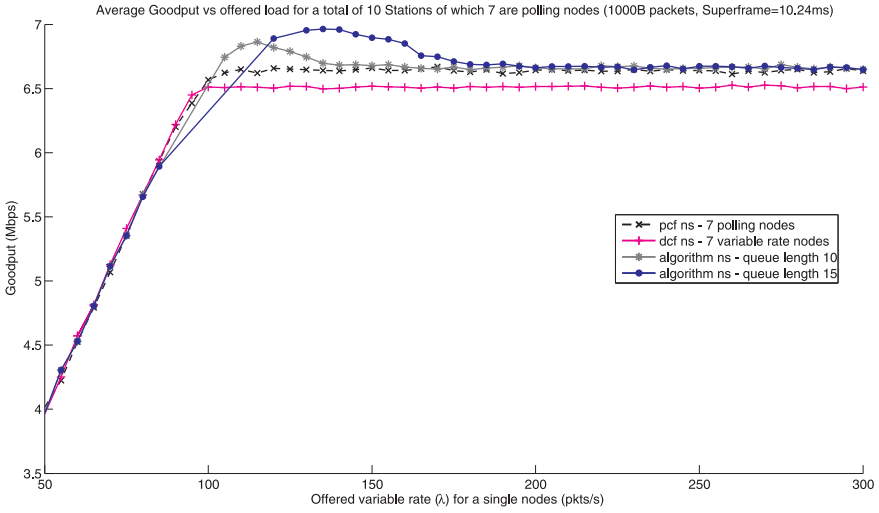


Figure 9.9: Average Goodput versus the packet arrival rate for a single node where the queue length parameter for the proposed protocol is varied (10 nodes, of which 3 have a fixed rate of $\lambda_{slow} = 50$ and 1000B packet size)

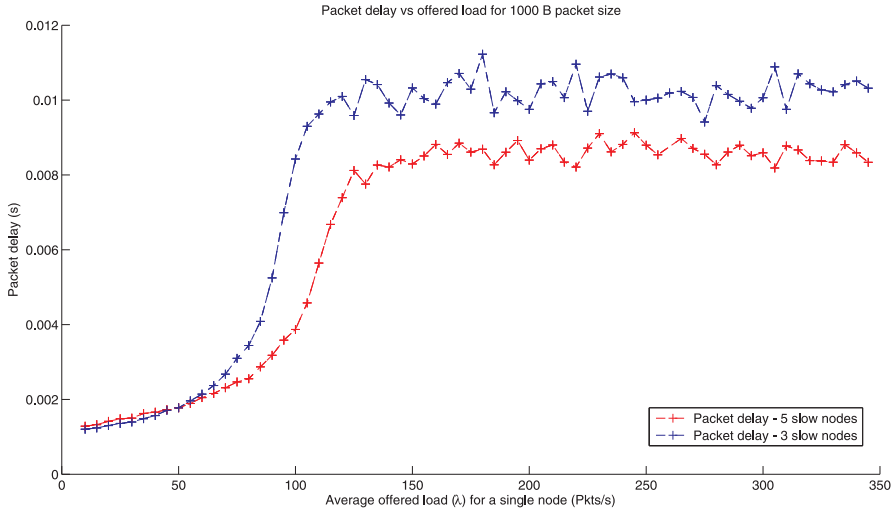


Figure 9.10: Comparison of the average DCF packet delay versus a variable packet arrival rate for a single node with a constant 1000B packet size (10 nodes, of which 3 and 5 have a fixed rate of $\lambda_{slow} = 50$)

Table 9.4: Result summary for Figure 9.10

Result type	Result
Average saturation DCF packet delay value for 10 nodes of which 5 transmit at rate 50 (pkts/s) and the other 5 at a variable packet rate	8.854 ms
Average saturation DCF packet delay value for 10 nodes of which 3 transmit at rate 50 (pkts/s) and the other 3 at a variable packet rate	11.267 ms

Table 9.5: Results summary for Figure 9.11

Criterion	Average saturation throughput simulation (Mbps)	Packet rate at which saturation starts (pkts/s)
DCF mode	6.487	85
2 polling nodes	6.663	90
3 polling nodes	6.744	90
5 polling nodes	6.985	100
7 polling nodes	7.149	115
Algorithm	7.049	110

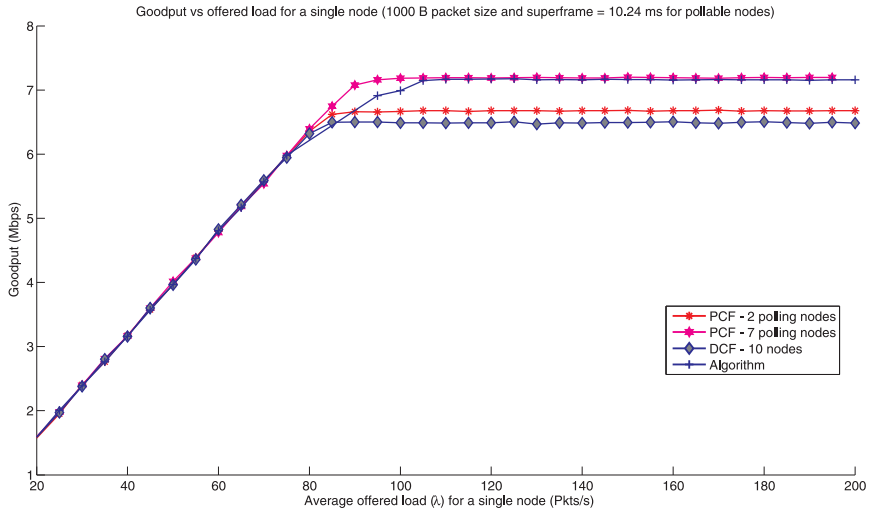


Figure 9.11: Comparison of the average goodput versus a variable packet arrival rate for a single node with a constant 1000B packet size (all 10 nodes transmit at the same rate)

9.3.4 Discussion of simulation results

As can be seen from figures 9.7, 9.8 and 9.9, the specified parameter checking the queue length to decide when the changeover from DCF to PCF should occur, has a significant effect on the goodput results. We define the initial overshoot as the values for which the algorithm's goodput is larger than the goodput obtained from the same setup for a PCF network. As saturation for normal PCF is reached, the algorithm exceeds this value to cause an initial overshoot, because some nodes are added to the polling list, which relieves some of the congestion for the nodes which have not yet been added to the polling list. Fewer nodes operating in the CP results in smaller packet delays and less buffer occupancy, implying that when saturation is reached some faster transmitting nodes are not added to the polling list, which results in a higher goodput. As the packet rate is further increased, the buffer occupancy increases for the faster transmitting nodes, and they eventually are added to the polling list, causing the algorithm to saturate to a network where 9 nodes are on the polling list and 1 node is in the CP.

Figure 9.9 is used to illustrate that there are times when the throughput obtained by the PCF mode of operation does exceed that of the DCF mode of operation, and specifically that the algorithm does harness this advantage. Figure 9.10 shows when PCF is better than DCF. When the average DCF packet delay increases beyond the value of the superfame, then PCF mode results in a higher throughput. From Table 9.4 for three slower transmitting nodes ($\lambda = 50 pkt/s$) and 7 variable rate nodes the average packet delay is $11.267ms$, whereas, for five fixed rate nodes ($\lambda = 50 pkt/s$) and five variable rate nodes the average DCF packet delay is $8.854ms$. The value for the superframe is $10.24ms$, thus $11.267ms$ exceeds the superframe value and the

PCF goodput will be higher.

Figure 9.11 illustrates when all stations transmit at the same rate the throughput is improved. The reason for this is that with the addition of stations to the polling list, due to their buffer occupancy exceeding the predefined queue length, the congestion in the CP is relieved. Stations left in the CP have a decrease in buffer occupancy as more stations are added to the polling list, until saturation operation is achieved.

Figure 9.11 is a comparison of certain protocols, where there is enough time provided for all nodes to be serviced in the CFP, because the superframe lasts 12.28ms, and the queue length parameter for the adaptive protocol is set to fourteen. It illustrates when all nodes are transmitting at the same rate with a small enough superframe, PCF outperforms DCF and, especially, the algorithm achieves improved throughput.

9.4 Proposed protocol buffer analysis approach

To predict the behaviour of the proposed protocol, an empirical approach will be used. The buffer occupancy for the scaled packet arrival rate DCF model will be used, to monitor the packet arrival rate at which the buffer occupancy will reach the predefined queue length. Using the mathematical model, the number of nodes on the polling list will be varied until the buffer occupancy goes below the predefined queue length value for a specific packet arrival rate.

In section 6.7.1 it was shown that for DCF nodes the buffer occupancy started increasing only once the throughput reached saturation.

9.4.1 Results

Table 9.6: Average buffer occupancy and packet arrival rates obtained from the mathematical model (for all fast nodes 1000B)

Average Buffer occupancy	for 1 CFP node	for 2 CFP nodes	for 3 CFP nodes	for 4 CFP nodes	for 5 CFP nodes	for 6 CFP nodes	for 7 CFP nodes	for 8 CFP nodes
1	77.972	82.383	80.969	82.562	84.444	86.786	88.977	92.191
2	81.499	86.146	84.507	87.034	89.661	93.585	98.328	106.31
3	82.421	87.256	85.652	88.256	91.869	96.164	101.84	111.86
4	83.312	88.367	86.797	89.478	92.901	97.51	103.88	115.07
5	84.204	89.477	87.936	90.616	93.933	98.787	105.41	117.37
6	85.096	90.671	89.044	91.648	95.005	99.888	106.76	119.26
7	85.987	91.956	90.152	92.68	96.165	101.01	108.06	121
8	87.634	93.241	91.26	93.896	97.324	102.35	109.46	122.7
9	89.357	94.723	92.979	95.506	98.921	103.83	111.01	124.47
10	91.319	96.676	94.799	97.31	100.67	105.56	112.78	126.4
11	93.899	98.768	97.212	99.527	102.82	107.7	114.88	128.56
12	96.95	101.42	100.03	102.31	105.5	110.27	117.4	131.12
13	100.66	104.56	103.55	105.71	108.81	113.55	120.55	134.21
14	105.47	108.38	108.08	110.11	113.13	117.69	124.59	138.15
15	111.88	113.27	114.15	116.03	118.94	123.39	130.1	143.43
16	121.23	119.78	123.18	124.91	127.59	131.87	138.33	151.32
17	137.42	129.64	138.88	140.46	142.93	146.96	153.01	165.28
18	178.67	146.9	179.7	181.48	182.96	186.45	NaN	201.25
19	NaN	199.59	NaN	NaN	NaN	NaN	NaN	NaN

Table 9.7: Average buffer occupancy and packet arrival rates obtained from the simulated model (for all fast nodes 1000B)

Average Buffer occupancy	For 1 CFP node	for 2 CFP nodes	for 3 CFP nodes	for 4 CFP nodes	for 5 CFP nodes	for 6 CFP nodes	for 7 CFP nodes	for 8 CFP nodes	for 9 CFP nodes
1	78.047	80.333	82.104	83.565	86.151	89.918	97.978	104.54	130.96
2	80.933	82.166	85.2	86.848	90.346	92.863	103.04	112.93	146.97
3	82.776	83.998	86.253	89.249	91.85	95.75	105.58	118.81	153.76
4	84.62	85.781	87.307	90.74	93.354	98.637	106.84	122.04	157.12
5	86.075	87.502	88.361	91.817	94.858	100.95	108.11	124.36	160.08
6	87.429	89.224	89.415	92.893	96.26	102.76	109.37	126.21	163.72
7	88.784	90.79	91.363	93.97	97.652	104.57	110.86	127.89	166.26
8	90.296	92.23	94.424	95.063	99.043	106.1	112.56	129.57	168.21
9	93.197	93.669	96.8	96.504	100.95	107.54	114.27	131.39	170.28
10	95.785	96.12	99.018	97.946	103.98	108.98	116.16	133.26	173.61
11	97.859	101.95	101.05	99.388	107.65	110.63	118.18	135.15	176.94
12	99.934	104.59	102.92	108.97	111.27	112.79	120.56	137.51	180.3
13	105.42	110.02	104.8	107.92	114.38	114.95	125.75	139.86	184.04
14	113.02	112.73	114.05	115.34	118.55	122.77	129.1	142.79	187.5
15	113.04	123.56	121.54	122.29	124.76	129.28	133.55	150.07	195.2
16	127.71	129.13	132.02	134.37	135.37	136.14	142.38	153.38	203.42
17	141.59	150.1	148.32	144.71	151.28	153.54	161.56	171.34	225.35
18	175.08	183.28	173.19	173.37	184.92	191.79	199.15	210.61	273.76
19	271.1	265.32	280.87	278.78	291.13	NaN	NaN	NaN	NaN

Table 9.8: Average throughput occupancy and packet arrival rates obtained from the mathematical model (for all fast nodes 1000B)

Average Buffer occancy	for 1 CFP node	for 2 CFP nodes	for 3 CFP nodes	for 4 CFP nodes	for 5 CFP nodes	for 6 CFP nodes	for 7 CFP nodes
1	6.2318	6.5368	6.4561	6.5714	6.6058	6.6928	6.6963
2	6.3885	6.7651	6.6812	6.7365	6.8573	6.9168	6.9123
3	6.3908	6.7875	6.6812	6.7618	6.8857	6.9644	7.0083
4	6.3931	6.8098	6.6812	6.7872	6.8999	6.9956	7.0563
5	6.3954	6.8322	6.6812	6.8126	6.9141	7.0112	7.0803
6	6.3977	6.8545	6.6812	6.8126	6.9283	7.0268	7.0894
7	6.3977	6.8545	6.6812	6.8126	6.9283	7.0268	7.094
8	6.3977	6.8545	6.6812	6.8126	6.9283	7.0268	7.0985
9	6.3977	6.8545	6.6812	6.8126	6.9283	7.0268	7.1031
10	6.3977	6.8545	6.6812	6.8126	6.9283	7.0268	7.1031
11	6.3977	6.8545	6.6812	6.8126	6.9283	7.0268	7.1031
12	6.3977	6.8545	6.6812	6.8126	6.9283	7.0268	7.1031
13	6.3977	6.8545	6.6812	6.8126	6.9283	7.0268	7.1031
14	6.3977	6.8545	6.6812	6.8126	6.9283	7.0268	7.1031
15	6.3977	6.8545	6.6812	6.8126	6.9283	7.0268	7.1031
16	6.3977	6.8545	6.6812	6.8126	6.9283	7.0268	7.1031
17	6.3977	6.8545	6.6812	6.8126	6.9283	7.0268	7.1031
18	6.3977	6.8545	6.6812	6.8126	6.9283	7.0268	NaN
19		6.8545	NaN	NaN	NaN	NaN	NaN

Table 9.9: Average throughput occupancy and packet arrival rates obtained from simulation (for all fast nodes 1000B)

Average Buffer occupancy	for 2 CFP nodes	for 3 CFP nodes	for 4 CFP nodes	for 5 CFP nodes	for 6 CFP nodes	for 7 CFP nodes	for 8 CFP nodes	for 9 CFP nodes
1	6.3471	6.4651	6.5858	6.6439	6.7785	6.8774	6.8838	6.8716
2	6.4596	6.6811	6.7147	6.8062	6.8521	6.986	6.9964	6.9754
3	6.572	6.6963	6.7918	6.8669	6.9258	7.0494	7.083	7.0157
4	6.6324	6.7115	6.8399	6.8973	6.9994	7.0649	7.1152	7.0416
5	6.6406	6.7267	6.8495	6.958	7.0315	7.0804	7.1383	7.069
6	6.6448	6.7419	6.859	6.9633	7.0467	7.0958	7.1554	7.095
7	6.6498	6.7587	6.8685	6.9739	7.0618	7.1185	7.1666	7.1067
8	6.6507	6.7635	6.878	6.9792	7.064	7.1327	7.1778	7.1171
9	6.6525	6.7611	6.8772	6.9839	7.0686	7.1398	7.1814	7.1276
10	6.6555	6.7572	6.8768	6.982	7.0709	7.1476	7.1886	7.137
11	6.6601	6.7604	6.8764	6.98	7.0722	7.1488	7.1958	7.144
12	6.6548	6.7707	6.8892	6.9808	7.0703	7.1497	7.1989	7.1511
13	6.6655	6.781	6.8896	6.9858	7.0684	7.1467	7.201	7.1492
14	6.6521	6.7765	6.8849	6.984	7.0739	7.1429	7.2009	7.1543
15	6.6598	6.7806	6.8805	6.9966	7.0715	7.1475	7.1989	7.1577
16	6.6597	6.7802	6.8875	6.9833	7.0771	7.1468	7.1996	7.1583
17	6.6614	6.7785	6.886	6.9768	7.082	7.1484	7.2	7.1562
18	6.6649	6.776	6.8889	6.979	7.0777	7.1494	7.1998	7.1592
19	6.6635	6.7818	6.8748	6.9839	NaN	NaN	NaN	NaN

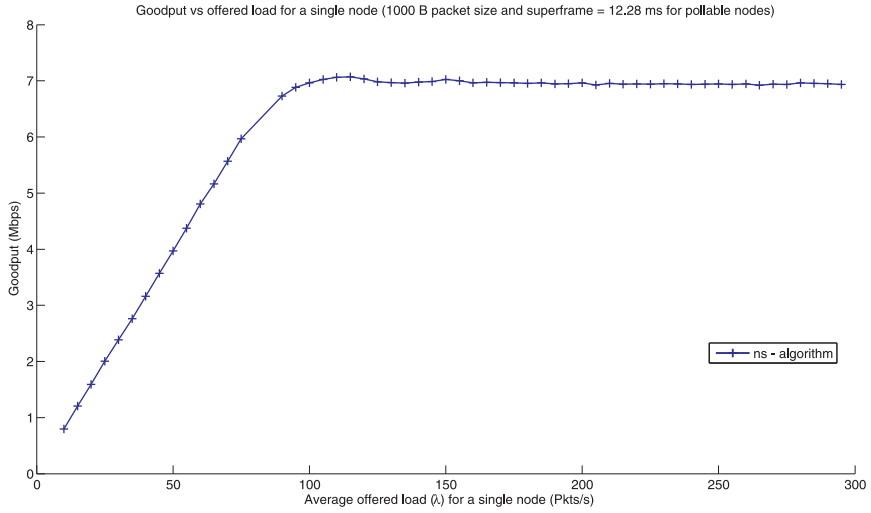


Figure 9.12: Average goodput versus the packet arrival rate of a single node (queue length parameter is 14, superframe=12.28 and constant 1000B packet size)

Table 9.10: Results for Figure 9.12

Packet arrival rate	Goodput (Mbps)
75	5.9686
80	6.2178
85	6.4728
90	6.7278
95	6.8825
100	6.9619
105	7.0242
110	7.0618
115	7.072
120	7.0353
125	6.9801
130	6.966
135	6.958
140	6.9799
145	6.9852
150	7.0231
155	6.9993
160	6.9633
165	6.9709
170	6.9694
175	6.9646

9.4.2 Discussion

The example illustrated here is for 10 nodes, all operating at the same packet arrival rate with a constant 1000B packets size, and $T_{SF} = 12.28ms$. The approach used can be applied to other similar network scenarios to predict the performance of the algorithm. It is known from sections 6.6.1 and 6.7.1 that the buffer occupancy only starts increasing once the system has reached its saturation throughput value which also implies the packet delay has saturated and reached its maximum value.

From the tables above it will be shown empirically how the algorithm handles a network of 10 nodes, all of which transmit at the same rate, uses the algorithm and which has a queue length parameter of 14. From tables 9.6 and 9.7 the results indicate an increase in the packet arrival rate for the queue length of 14 when more nodes operate in the CFP and fewer nodes in the CP. An explanation for this occurrence is that every time a node is added to the polling list, the service cycle time increases and the duration of the CP decreases. Fewer nodes contending for the channel in the CP will result in a decrease of packet delay but, the packet delay is increased by passing through a longer CFP, which does not outweigh the reduced packet delay of fewer nodes. By Little's law, an increase in the packet delay will require an increase in the packet arrival rate to obtain a similar buffer occupancy. In terms of the algorithm, every time a new node is added to the polling list and one removed from the CP, the packet arrival rate at which the same buffer occupancy occurs is increased.

Figure 9.12 illustrates the performance of the algorithm under scrutiny and Table 9.10 provides a summary of the results. The method used to predict the algorithm's performance was to refer to Tables 9.6 and 9.7, by looking at what packet rate the queue length falls below the required limit of 14. For example, from Table 9.10, the throughput for $\lambda = 120$ is 7.0353 Mbps obtained by ns2. Tables 9.6 and 9.7 are in agreement that the queue length is less than 14 for 7 CFP nodes. Table 9.6 predicts by method of mathematical modelling the average queue length for 3 nodes operating in the CP to be 13. Table 9.7 predicts by method of simulation the average queue length to be 12. Referring to Tables 9.8 and 9.9, which indicate the throughput for the mathematical and simulation approaches respectively to be 7.1031 and 7.1467 Mbps.

Another example is for $\lambda = 75$, both tables predict there are no CFP nodes operating and only CP nodes, which gives the throughput of 5.9686 Mbps. The last example is when the system reaches saturation, which implies the throughput reaches a constant value. In our example this is 6.9622 Mbps. The empirical approach predicts there will be 9 nodes operating in the CFP and 1 node in the CP. This can be predicted by the mathematical model as follows, 9 nodes operating in the CFP with a constant packet size of 1000B and a superframe that lasts 12.28 ms, and a utilization of one gives a throughput of 5.8632 Mbps (using equation 7.3.1). The

service cycle time for 9 nodes is predicted as 9.7818 ms (using equation 7.2.2). This gives the time available for transmission as 2.4582 ms. A packet requires 1.021 ms to be successfully transmitted, which implies roughly $2.4582/1.021 \approx 2$ packets are transmitted in each superframe, and 1/0.01228=81.433 superframes per second. This gives the packet per second for the single CP node as $81.433 \times 2.4076 \approx 162$, which gives the saturation throughput as $162 \times 8000 = 1.296$ Mbps. This predicts a saturation throughput of 7.1592 Mbps, which is in agreement with the simulation result obtained.

9.5 Summary

In this section the tradeoffs between PCF and DCF were illustrated. A method was also presented to illustrate the principle of deciding when PCF outweighs DCF. A proposed protocol was also designed, which monitors the queue length to notify the AP when changeover from DCF to PCF should occur.

Chapter 10

Summary and conclusions

In concluding this thesis the following remarks and observations are thought to be applicable.

10.1 Motivation

The IEEE 802.11 standard specified the Point Coordination Function as the deterministic protocol. Recently research into this aspect has stagnated, and it was the purpose of this project to investigate how existing infrastructure networks could be improved by optimising some modes of the 802.11 protocol. The investigation also hoped to determine when to change between DCF and PCF, and to provide an adaptive protocol to do so.

10.2 Summary of objectives

The following were the objectives of the thesis:

- First, to research the contention (DCF) and deterministic (PCF) protocols of the IEEE 802.11 standard.
- Thereafter to identify a software simulation package with which both PCF and DCF can be simulated.
- To make the necessary changes to the simulation software to enable the study of mathematical models, with which results can be compared.
- To investigate existing mathematical models and then combining them to predict the behaviour of PCF and DCF.
- To investigate under which circumstances PCF and DCF each perform better.
- To suggest a protocol that harnesses the advantages of DCF and PCF, and can dynamically adapt between them.

- It was not an objective of this project to identify a protocol to address the choice of size of the superframe repetition interval.

10.3 Summary of thesis

The following serves as a summary of this thesis:

- First an investigative study was made into contention (CSMA) and centrally scheduled (polling) protocols to provide insight into the workings of the IEEE 802.11 standard. This investigation also aided in the selection of a centrally scheduled protocol for the PCF mode of operation and to better explain the results obtained through simulation.
- An in depth study was made of the IEEE 802.11 standard, which helped to assure that all simulation and mathematical modelling was in agreement with the standards and specifications for accuracy.
- The building of the *ns*2 simulation to model an IEEE 802.11 network is described in detail, including the necessary configuration parameters. This included all discussions of changes made to the code to serve the purposes of this project.
- An in depth investigation was made into existing mathematical models which describe the DCF working. The most suitable model identified was that of Garetto, which could model the throughput, buffer occupancy, empty packet queue and the conditional collision probabilities.
- A model was identified by Chatzimisios that could be used with Garetto's model to obtain the DCF packet delay.
- An in depth investigation was made into the mathematical modelling of nodes operating in PCF mode. This included the packet delay, buffer occupancy and throughput.
- The models for PCF and DCF were combined by methods devised for this project to obtain the combined results for nodes operating the CFP and CP.
- Lastly, an in depth investigation was made into the traffic circumstances under which PCF and DCF modes provide the best performance for various performance metrics. From the results of this study a proposed protocol was presented.

10.4 Contributions of thesis

The following can be summarised as the contributions of this thesis:

- The establishment of a mathematical model that models all the regions of operation of an IEEE 802.11 b network. The model is predominantly based on existing mathematical models, but mathematical methods were contributed in combining a number of these models to provide one model to model the buffer occupancy, throughput and packet delay.
- An extensive working simulation was created, including changes made to the original *ns*2 code to obtain the necessary statistics and the proper functioning of the PCF toolbox for PCF nodes to operate only in the CFP, using round robin polling, as well as all the code for the mathematical modelling.
- A proposed protocol was subsequently defined that should be useful for deciding when to use either PCF or DCF, and a method of dynamically switching between them. The implementation of the protocol into *ns*2 is also included, which could serve as basis for real world implementation.
- A study of the circumstances under each of the two protocols, PCF or DCF, provides better performance is presented,to enable future decisions in network setup and throughput calculation.

10.5 Suggestions for future work

The investigation undertaken in this thesis is by no means complete and the following is a recommendation for possible future work:

- The problem of addressing the optimal superframe repetition interval presents significant research opportunities. This will include identifying the network circumstances for which certain values will be optimal and possibly to provide an adaptive algorithm.
- Further refinement of the mathematical model provided in this thesis, especially the packet delay.
- Investigating different polling algorithms that could provide improvements in performance.
- The proposed protocol provided has, through simulation, been proven viable. The next step would be to implement it in hardware for testing.

10.6 Final comment

The IEEE 802.11 standard has deeply embedded itself into wireless communications, and proved itself to be a reliable technology. With the increased scope for communications, and the resulting increase in the demand for bandwidth in a wireless finite usable spectrum, the need for optimising bandwidth usage is imperative.

This thesis attempts to make a contribution in this regard - by providing mathematical modelling and simulation of existing networks and as such, aiding in bandwidth optimisation.

Appendices

Appendices

All appendixes and code for simulations, mathematical modelling and results are on the accompanied CD.

Bibliography

[ABCPRV-05] P. Raptis, V. Vistas, K. Paparrizos, P. Chatzimisios, A. C. Boucouvalas, and P. Adamidis
Packet delay modeling of IEEE 802.11 Wireless LANs
in Proc. Intl. Conf. on Cyber. Tech. Sys. Apps (CITSA), July 2005

[BCV-03] P. Chatzimisios, A. C. Boucouvalas and V.Vitsas
IEEE 802.11 Packet Delay - A Finite Retry Limit Analysis
IEEE GLOBECOM, vol. 2, pp.950-954, December 2003

[Bia-00] G. Bianchi
Performance Analysis of the IEEE 802.11 Distributed Coordination Function
IEEE Journal on Selected Areas in Communications, Vol. 18, No. 3, pp. 535-547, March 2000

[Black-93] U. D. Black
Communications and Distributed Networks
Prentice Hall, 1993

[CG-05] M. Garetto, C.F. Chiasserini
Performance Analysis of the 802.11 Distributed Coordination Function under Sporadic Traffic
Networking 2005, Waterloo, Canada.

[CPW-02] H. Wu, Y. Peng, K. Long, S. Cheng, J. Ma
Performance of Reliable Transport Protocol over IEEE 802.11 Wireless LAN: Analysis And Enhancement
IEEE INFOCOM, 2002

[DM-07] D. Malone, K. Duffy, and D.J. Leith
Modelling the 802.11 distributed coordination function in non-saturated heterogeneous conditions
IEEE-ACM Trans. On Networking, vol. 15, No. 1, pp. 159-172, Feb. 2007

[FM-99] K.D. Fink, J.H. Matthews
Numerical methods using Matlab 3rd edition
Prentice Hall, 1999

[FV-08] K. Fall, K. Varadhan
The ns Manual
The VINT project, 2008

[Gast-02] M. Gast
802.11 Wireless Networks: The Definitive Guide, 2nd edition
O'Reilly Media, 2005

[[Hoc-97] N.C. Hock
Queueing Modelling Fundamentals
John Wiley & Sons, 1997

[Klein-75a] L. Kleinrock and F.A. Tobagi
Packet Switching in Radio Channels: Part I - Carrier Sense Multiple-access Modes and Their Throughput-delay Characteristics
IEEE Transactions on Communications, Vol. COM-23, No. 12, December 1975

[Klein-75b] L. Kleinrock and F.A. Tobagi
Packet Switching in a Multi-access Broadcast Channel: Performance Evaluation
IEEE Transactions on Communications, Vol. COM-23, No. 4, April 1975

[Klein-75c] L. Kleinrock and F.A. Tobagi
Random Access Techniques for Data Transmission over Packet-switched Radio Channels
Proceedings, National Computer Conference, 1975

[Lam-75] S.S. Lam and L. Kleinrock
Packet Switching in a Multi-access Broadcast Channel: Dynamic Control Procedures
IEEE Transactions on Communications, Vol. COM-23, September 1975

[Maral-99] G. Maral and M. Bousquet
Satellite communications
John Wiley & Sons, Ltd, November 1999

[Sikdar-05] B. Sikdar
Delay Analysis of IEEE 802.11 PCF MAC based Wireless Networks
Proceedings of IEEE GLOBECOM, November 2005

[ST-04] O. Tickoo and B. Sikdar
Queueing analysis and delay mitigation in IEEE 802.11 random access MAC based wireless networks
Proceedings of IEEE INFOCOM, pp. 1404-1413, Hong Kong, China, March 2004

[SV-04] I. Vukovic and N. Smavatkul
Delay Analysis of Different Backoff Algorithms in IEEE 802.11
Proc. Of IEEE Vehicular Technology Conference (VTC), Los Angeles CA, September 2004

[Tobag-80] F.A. Tobagi
Multi-access Protocols in Packet Communication Systems
IEEE Transactions on Communications, Vol. COM-28, No. 4, April 1980

[Wolhu-03] R. Wolhuter and G.J. van Rooyen
Elements of Telecommunications Systems Design and Teletraffic Analysis
Stellenbosch, 2003